AD-A096 317   WESTINGHOUSE DEFENSE AND ELECTRONIC SYSTEMS CENTER B--ETC F/G 17/5

INTELLIGENT TRACKING TECHNIQUES.(U)

OCT 80 T J WILLETT, J ABRUZZO, V ZAGARDO   DAAK70-78-C-0167

UNCLASSIFIED   NL

1 OF 1

AD A
096317

END
DATE
FILMED
4-81
DTIC

LEVEL III

3

AD A096317

INTELLIGENT TRACKING TECHNIQUES

FIFTH QUARTERLY REPORT

For

DECEMBER 31, 1979

CONTRACT: DAAK 70-78-C-0167

Presented to

Night Vision and Electro-Optics Laboratory
Fort Belvoir, Virginia 22060

Submitted by

Westinghouse Electric Corporation
Systems Development Division
Baltimore, Maryland 21203

INTELLIGENT TRACKING TECHNIQUES

FIFTH QUARTERLY REPORT

For

DECEMBER 31, 1979

CONTRACT: DAAK 70-78-C-0167

Presented to

Night Vision and Electro-Optics Laboratory
Fort Belvoir, Virginia 22060

Submitted by

Westinghouse Electric Corporation
Systems Development Division
Baltimore, Maryland 21203

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A096317	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Intelligent Tracking Techniques, Fifth Quarterly Report		5. TYPE OF REPORT & PERIOD COVERED Fifth Quarterly Report October 1,1979-Dec. 31,1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) T. J. Willett et. al.		8. CONTRACT OR GRANT NUMBER(s) DAAK 70-78-C-0167
9. PERFORMING ORGANIZATION NAME AND ADDRESS Systems Development Division Westinghouse Electric Corporation Baltimore, Maryland 21203		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Night Vision and Electro-Optics Laboratory Fort Belvoir, Virginia 22060		12. REPORT DATE Oct. 28, 1980
		13. NUMBER OF PAGES 85
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automatic Target Cueing	TV Sensor
Target Recognition	Digital Image Processing
Target Tracking	Correlation Tracking
FLIR Sensor	Target Homing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is the Fifth Quarterly Report under a contract to investigate the design, test, and implementation of a set of algorithms to perform intelligent tracking and intelligent homing on FLIR and TV imagery. The intelligent tracker will monitor the entire field of view, detect and classify targets, perform multiple target tracking, and predict changes in target signature prior to the targets entry into an obscuration. The intelligent tracking and homing system will also perform target prioritization and critical aimpoint selection.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The system concept was described. The problem of target aspect determination in support of aimpoint selection was analyzed. Sequences of 875 line FLIR data were extracted from the NV&EOL data base and an example of aspect determination for a maneuvering target in the presence of obscurations was presented. An example was also presented for close-in homing (less than 500 meters) and the emergence of interior features, target movement, and scale changes. Hardware implementation in terms of VLSI/VHSIC chips was analyzed.

A

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

INTRODUCTION

Under contract to the Army's Night Vision and Electro-Optics Laboratory. Westinghouse has been investigating the design, test, and implementation of a set of algorithms to perform intelligent tracking and intelligent target homing on FLIR and TV imagery. Research has been initiated for the development of an intelligent target tracking and homing system which will combine target cueing, target signature prediction, and target tracking techniques for near zero break lock performance The intelligent tracker will monitor the entire field of view, detect and classify targets, perform multiple target ttracking, and predict changes in target signature prior to the target's entry into an obscuration. The intelligent tracking and himing system will also perform target prioritization and critical aimpoint selection. Through the use of VLSI/VHSIC techniques the intelligent tracker (with inherent target cuer) can be applied to the fully-autonomous munition.

During the fifth quarter. several meetings and a number of phone conversations took place between Westinghouse personnel and CPT. Reischer of NV&EOL. The system concept was described. The problem of target aspect determination in support of aimpoint selection was analyzed. Sets of 875 line FLIR imagery were extracted from the NV&EOL data base and an example of aspect determination for a maneuvering target in the presence of obscurations was presented. An example was also examined for close-in homing (less than 500 meters) and the emergence of interior features, target movement, and scale changes. Hardware implementation in terms of VLSI/VHSIC chips was also analyzed.

Westinghouse personnel participating in this effort include Thomas Willett, Program Manager, Joseph Abruzzo, Vincent Zagardo, John Shipley, Leo Kossa, and Richard Kroupa. Program review and consultation is provided by Drs. Glenn Tisdale and Azriel Rosenfeld.

1.0 SYSTEM CONCEPT

The purpose of this section is to describe the intelligent tracker concept that has evolved during the course of this work. This Fifth Quarterly Report marks the end of the analysis and design phase; hence, the system concept outlined below will be employed during the test phase. For the test phase, to be conducted during the next quarter, several video tape sequences will be prepared to exercise the intelligent tracker functions and the system concept. The desired intelligent tracker functions are:

1. acquisition and track initiation - detect, locate, classify and prioritize targets automatically and hand off to the internal tracker (the intelligent tracker concept is assumed to include both acquisition and tracking);
2. handle multiple targets - track a number of targets in a scene simultaneously;
3. target signature prediction predict or anticipate target obscuration and how the target signature will change as a result of the obscuration;
4. reacquisition - reacquire a target at the earliest opportunity following track breaklock, including departure from the field of view;
5. aimpoint selection - determine the critical aimpoint of a target (which may be an interior point within its silhouette).

A video timing diagram for the intelligent tracker is shown in Figure 1-1. Figure 1-2 shows how an image would appear in the frame memory. A block diagram of the system concept is shown in Figure 1-3.

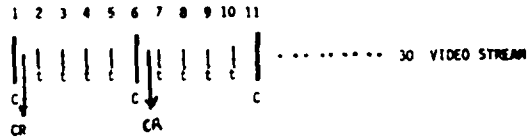


Figure 1-1. Timing Diagram

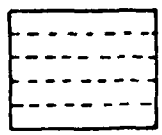


Figure 1-2. Image in Frame Store Ram

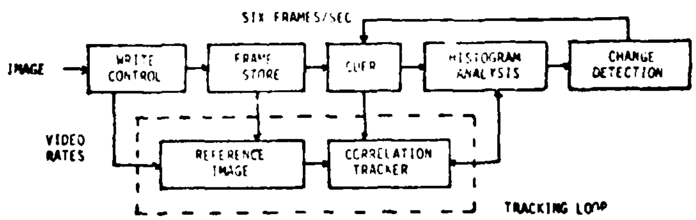


Figure 1-3. System Block Diagram

The heavy vertical lines in Figure 1-1 represent the cued frames and the lighter vertical lines are the tracked frames. Frame 1 is a cueing frame; frames 2, 3, 4, and 5 are tracking frames, and then the cycle repeats. Frame 6 is a cueing frame, and frames 7, 8, 9, and 10 are tracking frames. Thus, the complex target cueing function is performed at one-fifth the rate of the simpler tracking function.

Figure 1-2 represents the image in the frame store which is divided into five horizontal and addressable strips. This is done so that, instead of waiting for the entire frame to be cued before handover to a tracker, targets can be handed over more rapidly. The reduction in handover lag increases the confidence that both cuer and tracker are working on the same target, and reduces the size of the track window. The frame store RAM is organized into five independently addressable strips to aid in sending the gray levels surrounding the target to the tracker for a reference image and to decrease the access time. It should be pointed out that the five tracked frames per cued frame and five strips for frame storage are initial estimates serving as a strawman concept. The point of this discussion is that the cuer results, labelled CR in Figure 1-1, can be utilized between the first and second frames instead of waiting for the fifth and sixth frames in the video stream.

Referring to Figure 1-3, we describe the intelligent tracker functions previously listed:

1.1 TARGET ACQUISITION AND HANDOVER

A horizontal strip of a frame is snatched in real time and placed in a frame memory. The cuer processes the strip and detects and classifies all targets. That part of the frame store holding each target and its surrounding gray scale window is sent to the tracker as a reference image. The tracker converts this to a binary reference image and now tracks this target from the video stream until the next cued frame. The cuer tells the tracker where the target is within the frame so the tracker can tell the write control when to write the next target window (now from the video stream) and subse- quent windows into its reference image.

1.2 MULTIPLE TARGET TRACKING

There is a reference image RAM reserved for each target. Only one correlator may be required, since it is fast enough to be multiplexed among 15 targets per frame (Second Quarterly, page 1-9). The tracker is a bandpass binary correlation tracker in this preliminary design and the bandpass (band of gray levels representing the target) is adjusted at the cueing rate by the cuer. Additionally, the tracker forms a smoothed track for each target using second order difference equations (to interface with the rate loop in the sensor gimbal). This allows reacquisition of a target which has left the field of view but can be brought back into view by moving the sensor along the image centered target track.

At this point one can picture simultaneous multitarget cueing and tracking. The cued targets have been classified and are ordered in an internal table in terms of priority. The priority precedence has been determined before the mission and loaded in the cuer and may be updated as conditions change during the mission.

1.3 TARGET SIGNATURE PREDICTION

To predict obscurations, a histogram of the background ahead of the target is analyzed. The track window errors, which are used to form the smoothed track for reacquisition, are also used to determine the position of the histogram window in front of the target. From the histogram, we can compare the gray levels ahead of the target with those of the target. If the same gray levels are present in both, a clean target segmentation is unlikely. The track point position is adjusted within the track reference window so that the tracker is using that portion of the target which will be obscured last (e.g. the rear of a target passing behind some low lying shrubs). The background histogram is also analyzed for a polarity reversal between the target and the background. An example of this is where the target is moving from a dark background into one lighter than itself and hence becomes a dark target against a light background. Under this condition, the new background is segmented and binary change detection at the background level is used along with direct seg- mentation to detect the target. Once having found the emerging edge of the target, the track point in the reference is adjusted to the front or emerging edge.

1.4 REACQUISITION

The reacquisition function comes into play when track breaklock occurs. It is handled by a combination of cueing, histogram analysis, and change detection. A difficult problem is the reappearance of a target which is partially obscured. Segmentation of the entire shape is not possible, which impedes automatic target recognition. In this case, scene change detection provides one of the few cues for target reemergence. Performing a histogram analysis of any changed areas segmented adds useful information because, in some cases, the target histogram will exhibit a peakedness not found in a background object such as a woods clearing. That is, the histogram will resemble a Gaussian rather than a Uniform Distribution. Otherwise, the clearing may be mistaken for a partially occluded target.

1.5 AIMPOINT SELECTION

Aimpoint selection is the process of designating a vulnerable point on the target, aiming the sensor at that point, and homing on it in the case of a missile. Target classification and target aspect are inputs to the aimpoint selection process. Target classification is available to the intelligent tracker as the synergistic result of including a cuer in the concept. Target aspect can be determined from target motion, from the orientation of the target edges in the image plane, and from the orientation of the principal axes. The direction of motion is used to determine the front of the target and to identify those target edges associated with the sides of the target, i.e. the orientation angles of those edges should be similar to the direction of motion. Analysis of the edge joining the "side edge" tends to indicate the orientation of the end of the target in the image plane. The orientation of the principal axes tends to indicate the bulk of the upper target mass (the turret in the case of a tank), and aids in target aspect determination. This approach was applied success- fully to a maneuvering target .

Given a classified target with aspect determined, the aimpoint is selected by comparing the target to one stored in memory with the approximate classification and aspect. The aimpoint is placed on the target through computations based on a similar triangle argument with respect to the stored image. Reference points are the centroid and the four intersections of the principal axes with the target silhouette for both the real and stored targets.

For a missile, homing at long range is controlled by the binary correlation tracker. For close in homing, 24 lines of target or more, homing is based on the successive registration of connected edges associated with the complete target image, exterior and interior. The hand-over from long range homing to close in homing is accomplished by simultaneous correlation and registration of the same target image. Tracking and homing are conducted at frame rates.

2.0 DETERMINATION OF AIMPOINT SELECTION INPUTS

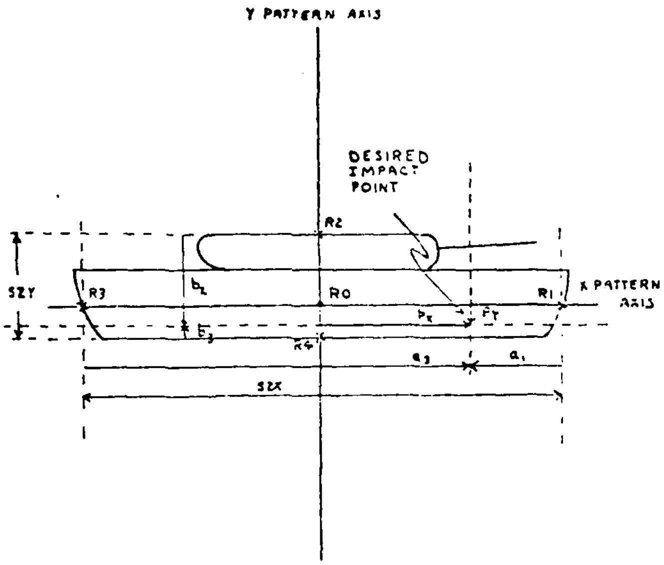
The aimpoint selection algorithms for the intelligent tracker were described in the Fourth Quarterly Report. The required inputs to aimpoint selection are target classification and target aspect. Determination of these inputs is the subject of this section. The aimpoint selection algorithms are briefly reviewed, the source of the target classification input is discussed, and target aspect determination is analyzed.

2.1 REVIEW OF AIMPOINT SELECTION ALGORITHMS

Prior to the mission, the aimpoint is determined by analyzing the target's vulnerable area and the most likely mission scenario(s). Once determined, the aimpoint information is stored in terms of a feature vector, F, for each target class and several aspects. The feature vector is constructed by forming a silhouette of the target for a typical aspect angle. The silhouettes can be divided between those typically seen by a low-flying, roll-stabilized weapon and those top views seen by a ballistic weapon with a near vertical descent. During the mission a target is detected, classified, and its aspect estimated. The aspect data is used to select a feature vector from memory for that class of target and an aimpoint is computed on the actual target.

Figure 2-1 shows the principal pattern axis drawn on a tank and the selected aimpoint. The offsets of the aimpoint from the four intersections of the target with the principal axis (R1-R4), as well as the perpendicular offsets from the pattern centroid (R0), define the aimpoint. These offsets, along with the major axes of the pattern, define the feature vector. Computations based on similar triangle arguments are used to transfer the aimpoint from the reference target to the actual target. Note that the centroid, principal axes, and principal axes-target intersection calculations are computed on the binary image which is used to form the reference image for the binary correlation tracker.

For a partially obscured targt with one or more of the edges (hence offsets) missing, the aimpoint computation is very similar. Assuming a constant orientation angle, the centroid correction, and accompanying offsets can be computed by estimating target dimensions before obscuration. On subsequent frames after the target has been obsucred, size estimates are generated and compared to the initial estimates. The observed, but erroneous, centroid measurement is then corrected on the basis of comparing the true and observed pattern sizes.



$\bar{F} = (SZX, SZY, a_1, b_2, a_3, b_4, P_x, P_y)$

Figure 2-1. Aimpoint Specification

2.2 DETERMINATION OF TARGET CLASS

A cuer is an integral part of the intelligent tracking and homing system, as described in Section 1.0; target class is an output of the cuer. The availability of target class is another example of the synergism achieved by combining a cuer and tracker into one system. Misclassification results in an erroneously placed aimpoint because the wrong target model is employed. For generality, it is assumed that the target classification algorithms contained in the cuer do not provide target aspect determination, which will be considered next.

2.3 TARGET ASPECT DETERMINATION

There are a number of interwoven issues involved in aimpoint selection and target aspect determination. These issues are discussed separately in the next several subsections. In the concluding subsection, their interrelationship is considered. The issues discussed in relation to target aspect determination are the allowable missile "g" forces, missile guidance settling times, detection and classification ranges, steering capability vs range, objectives of exterior aimpoint selection, missile trajectory and a partial determination of target aspect, complete target aspect determination, and simplification of the close-in homing problem.

2.3.1 Allowable Missile "g" Forces

In the Fourth Quarterly Report, a missile and its dynamic characteristics were assumed. The maximum turning rate was 3 g's. For a missile traveling at 500 meters per second, this corresponds to a maximum turning rate of 3.4 degrees/second. This means that the target aspect determination throughout the missile flight must be accurate enough to avoid requiring a corrective turn exceeding 3.4 degrees/second.

2.3.2 Missile Guidance Settling Times

From the Fourth Quarterly Report, the assumed missile settling time is one second from impact. At a speed of 500 meters per second, the aimpoint and underlying aspect determination solutions are essentially completed at a range of 500 meters from impact. The impact CEP was assumed to be one meter. For the staring sensor with a field of view of 0.24 radians and an angular resolution of 0.15 mrad, as assumed previously, approximately 30 lines will appear on the target at 500 meters. This means that the major course corrections for aimpoint selection must occur on a target smaller than 30 lines high. A possible ramification of this constraint is that the exterior aimpoint selection process must put the aimpoint within a one meter CEP of the desired impact. This desired aimpoint may be confirmed and slight steering adjustments initiated by an interior feature which is not yet visible. Note that the 30 line target subtends approximately 0.7^{o}; hence, slight adjustments (measured in pixels) may still be possible before impact.

2.3.3 Detection and Classification Ranges

Detection and Classification ranges were found to be 2700 and 1600 meters respectively, based upon work in the previous Quarterly Report. The missile could begin turning toward the detected targets at 2700 meters with the final target selection reserved for the classification range of 1600 meters. At this range the possible targets are limited to those within 6.8 degrees (2.0 seconds x 3.4 degrees/seconds) of the missile longitudinal axis since the missile has only two seconds flight time to close to within 500 meters of impact (the guidance settling range). The other factor, 3.4 degrees/second, is the maximum turning rate. Further, at a range of 1600 meters, target aspect determination can be estimated with knowledge of the target class. At this point, the target is only some 8-10 lines high, so there is a question of how accurately this initial aimpoint and underlying aspect determination needs to be calculated. This issue leads to the discussion in the next paragraph.

2.3.4 Performance of Exterior Based Aimpoint Selection

At classification ranges of 1000 meters and with a ten-line target, the aimpoint images shown in the previous Quarterly Report, Figure 2.1-16 (images 224, 228, ... 260) show a variability in aimpoint location of 1.0 to 2.0 pixels from frame to frame for the side view of a tank. At this range 1.0 pixel corresponds to 0.6 meters, so the variability circle has a radius of 1.2 meters. This compares favorably with an assumed aimpoint CEP of 1.0 meters. The variability should diminish as the range closes and the video signal-to-noise ratio of the exterior edges increases. This means that a serious attempt can be made at classification ranges to place the aimpoint correctly on the target. That is, the aimpoint tracking performance is good enough to go beyond just placing the aimpoint anywhere on the target and upgrading the accuracy as the range to impact diminishes further.

2.3.5 Missile Trajectory and Partial Target Aspect Determination

A dichotomy of aimpoint reference images based on low-flying and ballistic trajectories was mentioned in para. 2.1 and is discussed here. The idea is to use the missile trajectory and target classification information to compute the amount of target side and top area which appear in the target projection in the image plane. At classification ranges, it is difficult to distinguish between the side and top portions of the target image. In Figure 2-2, three views of a square target are seen which correspond to each of three trajectories. Trajectory 1, representing the low-flying case, sees only the side view of the target. Trajectory 3, representing a near-vertical, ballistic terminal phase sees only the top of the target.

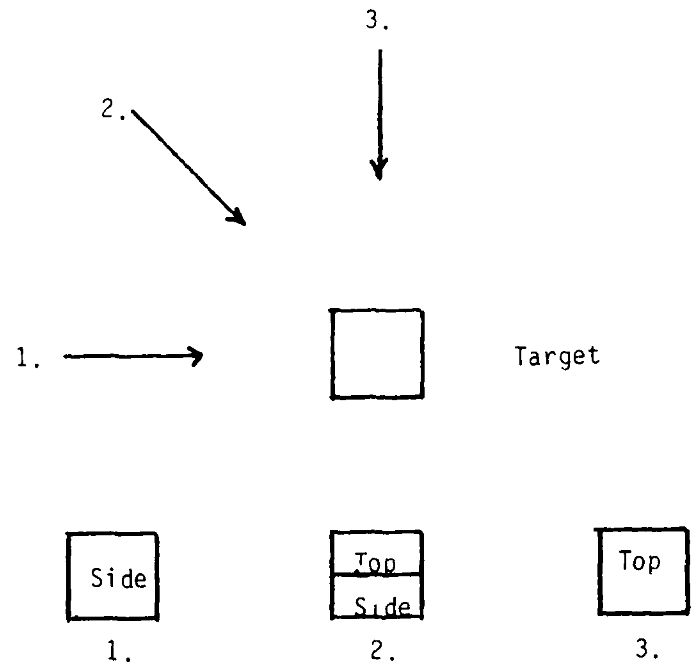


Figure 2-2. Three Trajectories and Their Respective Images

Trajectory 2, representing a 45 degree terminal descent sees both the side and top of the target in its image. For the discussion at hand, consider a low-flying missile at 3km ground range from the target at an altitude of 61.1 meters (200 feet) with a field of view of 0.24 radians, as shown in Figure 2-3. Assume the angular resolution of the sensor is 0.15 mrad and is equivalent to 0.5 minutes per line. The angle, θ_1, between the vertical and the base, A, of the target is $\theta_1 = \tan^{-1}(3000/61.12) = 88^{\circ}\ 50'$. The angle, θ_2, between the vertical and point B of the target is $\theta_2 = \tan^{-1}(3000/59.12) = 88^{\circ}\ 52'$. Note that this implies two minutes of arc are on the target or 4-5 lines which is in agreement with the previous work in the Fourth Quarterly Report which found 4-5 lines on the target at 3000 meters for the assumed sensor . The angle from the vertical to

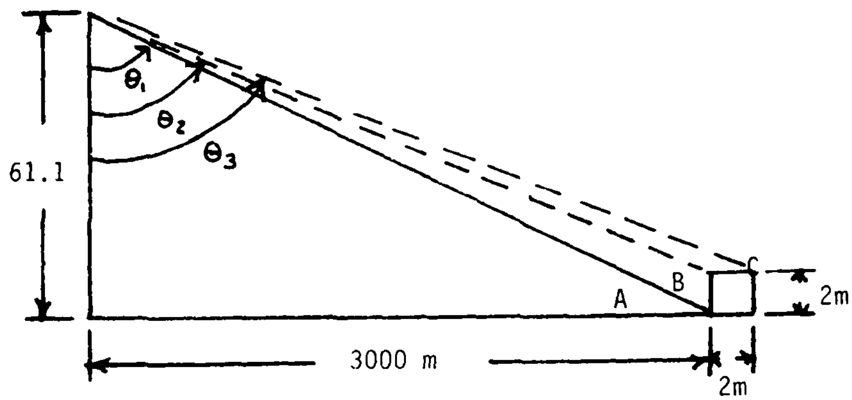


Figure 2-3. Geometry for Low-Flying Missile Case

point C is $\theta_3 = \tan^{-1}(3002/59.12) = 88^0\ 52.04'$ so the increase in angular coverage from point B to point C is less than 0.04 minutes and much less than an additional line. Hence, for the low-flying case in the vicinity of the detection range, the largest aspect is entirely a side view, and is Trajectory 1 in Figure 2-3.

Consider the low-flying missile closing to a classification range of 1500 meters, and an altitude of 61.1 meters (200 feet). The previous calculations can be repeated. Here, $\theta_1 = 87^0\ 40'$, $\theta_2 = 87^0\ 44'$, and $\theta_3 = 87^0\ 44.3'$. In this case, the missile sees a part of the top, but the maximum amount of top shown is no more than one line on a target which has eight lines of side projection. These calculations are significant because they provide a prior knowledge of the target aspect in one of the three directions. For example, given the target class, sensor resolution, and a low-flying trajectory, the number of lines on the target can be computed as a function of range. Further, the number of lines apportioned between the side and top can be computed as a function of range. Hence, at some point in the trajectory the measured number of lines on a target classified by the cuer can be divided between those representing the side and those representing the top. This determines the target aspect in a plane containing the missile and the target. The aspect angle is measured from the vertical to the line of sight as in θ_3 of Figure 2-3. Finally, this computation is important in determining the other two rotation angles for the remainder of the target aspect determination.

2.3.6 Development of 3-D Target Aspect Determination

The development of three-dimensional target aspect determination is presented in this section. The aspect which a ground target presents to an incoming missile is a function of the missile trajectory, the position of the missile in that trajectory, and the relative orientations of the ground target about three axes of rotation. To simplify the aspect discussion, consider the missile fixed in space at some arbitrary point and the ground target free to rotate completely about a set of mutually perpendicular, body centered axes which are fixed at the centroid of the target. By allowing the target to have complete rotation, not constrained by the ground, any target aspect can be reproduced with respect to the missile fixed in space. Figure 2-4 shows the three mutually perpendicular axes of rotation. The x and y

axes are in a plane containing them and the centroid. The z axis is positive outward. Rotation about the x axis produces the side versus top view discussed in the

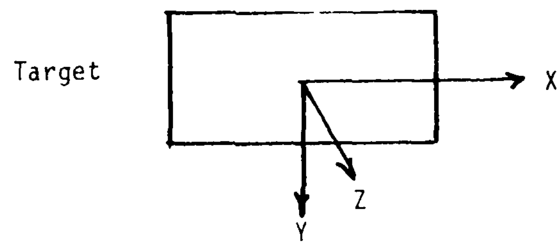


Figure 2.4 Mutually Perpendicular Axes of Rotation Fixed in the Target.

previous section and seen in Figure 2-5. The second rotation is about the y axis and produces a side-end composite view as shown in Figure 2-6. The third rotation is about the z axis and produces a view which shows the target on an incline as depicted in Figure 2-7.

From the discussion of Section 2.3-5, the angle of rotation, α , about the x axis has been computed from the missile trajectory, sensor resolution, and target classification. Consider the computation of β for the case of a rotation about the y axis as shown in Figure 2-6. Assume the target shown in Figure 2-6 is a tank which, referring to the Fourth Quarterly Report on page 2-13, is six meters long, 2.3 meters wide and 2.4 meters high. The tank dimensions along the x axis can vary from a minimum of 2.3 meters for an end view to 6.5 meters for a side view with 45^{o} rotation. There is a relation between the measured x dimension and the amount of rotation in the angle beta about the y axis. However, there are ambiguities in the relationship which are discussed by considering the rotation shown in Figure 2-8. There, two ambiguities arise from only knowing the x dimension. The angle of rotation can be 45^{o} or 135^{o}, and the front and rear of the target are also unknown. Under the assumption of a massive armor attack, as discussed in the Fourth Quarterly Report, the tanks will be moving rapidly. Hence, the direction of motion, as determined by the binary correlation tracker and the change detection algorithms described in the Third Quarterly Report, can be used to define the front and rear of the tank. Further analysis can be divided into an examination of interior shapes and exterior con- nected edges associated with the velocity vector.

To resolve the remaining ambiguity, consider the approach of using the cuer to segment interior tank features as the range closes. At this point in

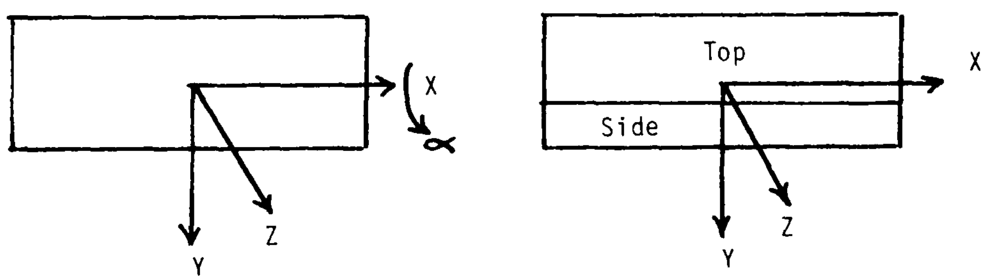


Figure 2-5. Rotation about the x Axis.

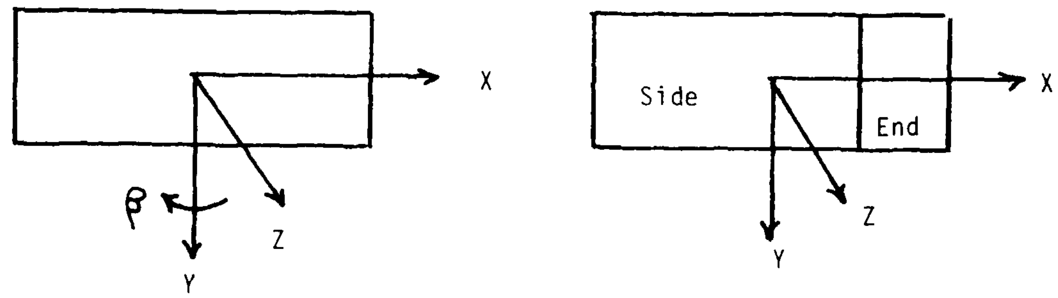


Figure 2-6. Rotation about y Axis.

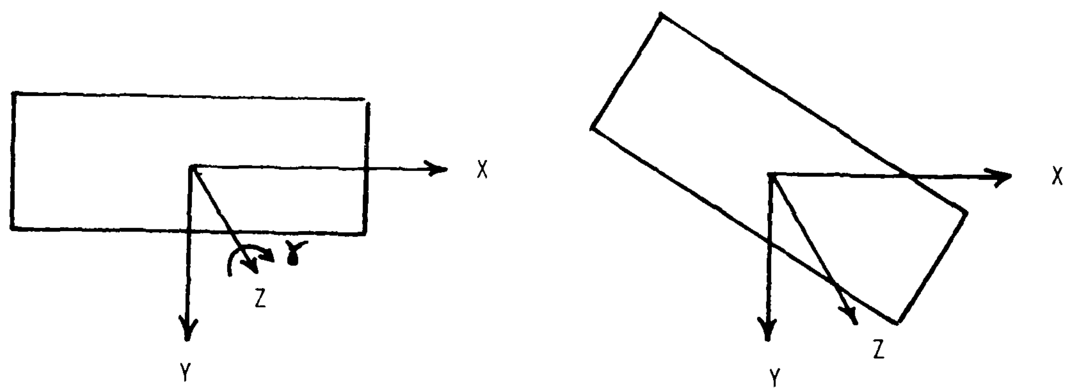


Figure 2-7. Rotation about the z Axis.

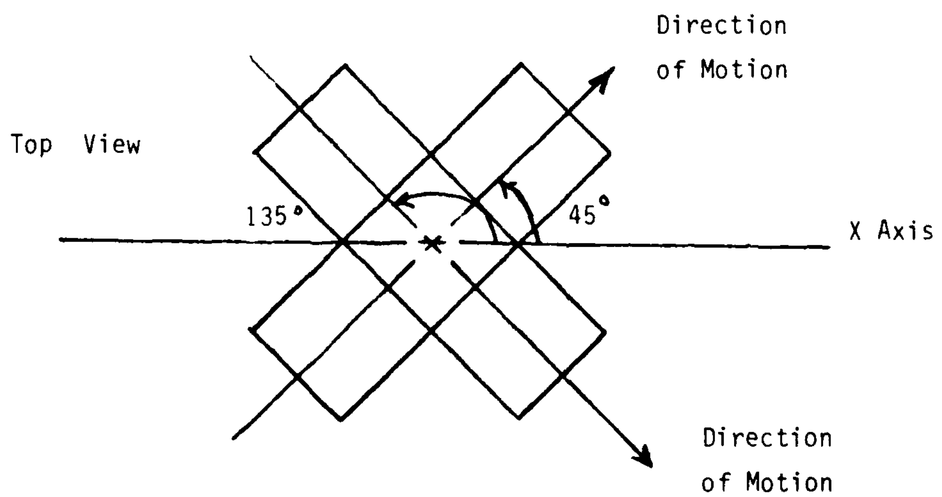


Figure 2-8. Two Rotations about the y Axis.

the analysis it is important to discuss the problem definition. The proposed approach is not to extract and classify a large set of features such as turrets, gun barrels and so on and then classify the target. Rather, the approach is to extract a small set of shapes which are known to exist on particular portions of the target that has already been classified. This is a significantly smaller and more tractable problem and will owe its success to the synergistic effect of combining a cuer and tracker. The set of interior shapes should be large enough so that they can be represented by a significant number of lines even at longer ranges. They should exhibit fairly high signal-to-noise ratios so that segmentation is straightforward. Finally, they should have a geometric position on the tank so that their detection would resolve the ambiguity shown in Figure 2-8. The engine hot spot is one such feature. It appears at the rear of the tank, can be represented by as many pixels as the width of the tank, and has a signal-to-noise ratio that can be 15 gray levels higher than the remainder of the tank. Additional features are the tank treads as seen in end or quartering views. The treads can be one-quarter of the width of the tank. A pair of treads on one side of a tank silhouette indicates an end on that side, and the signal-to-noise ratio can be seven gray levels higher than the remainder of the tank, excluding the engine hot spot. Using these features to resolve the ambiguity of Figure 2-8, the hot spot would be more apparent in the 45° rotation, and it might even obscure the treads which would appear on the left end of that target. The treads would be more apparent on the right end of the tank silhouette produced by the 135° rotation. In fact, in both cases the distance between treads is a rough measure of the amount of rotation. This approach requires a set of such features stored for each target type.

A complementary approach to resolve the remaining ambiguity is further analysis based on motion. A tracked vehicle such as a tank travels in the direction of its longitudinal axis. The longitudinal axis of a tank is parallel to its sides. Suppose the segmentation of a tank is composed of a set of connected edges for which the orientation angles in the image plane have been computed. From motion analysis, the orientation of the velocity vector in the image plane has been computed. By comparing the orientation angles of the velocity vector with those of the connected edges describing the target exterior, that set of target edges which approximate the velocity vector can be labelled as the side of the tank. Then the remainder of the edges plus the orientation of the principal axes may be used to "flesh out" the rest of the tank.

The final angle of rotation for target aspect determination is the angle γ about the outward z axis, as shown in Figure 2-7.

γ can be directly calculated in the image plane by computing the orientation angle of the lower or bottom edge defining the target shape. The procedure for computing orientation angles for edges was presented in the First Quarterly Report.

2.3.6.1 An Example of Target Aspect Determination

In the above section, the determination of each angle of rotation was considered separately. The angle determinations are now considered together in an example. In considering the motion of a tank, for example, it is noted that the longitudinal axis of the tank is generally parallel to the direction of motion. Further, the extracted lower edges describing the sides are parallel both to the motion and the longitudinal axis. Hence, as an a first approximation to the problem of deciding which rotation angle to determine first, the direction of motion is found. Then that target lower edge which is parallel to the direction of motion describes the side, as shown in Figure 2-9. A rough approximation to the end of the target can now be attempted. Important features are a pair of vertices shown at points A (`_ _) and B (_ _|) of Figure 2-9. Having detected these vertices, as shown in Figure 2-9 by dashed lines, the remainder then can be completed, shown by solid lines. From the 875-line FLIR data supplied by NV&EOL, it appears that information of this type is discernable on target of 12 lines high. Now the angles of rotation can be computed. The direction of motion and identification of the lower, parallel edge suggest which side (left or right) will contain the end view and the A and B vertices.

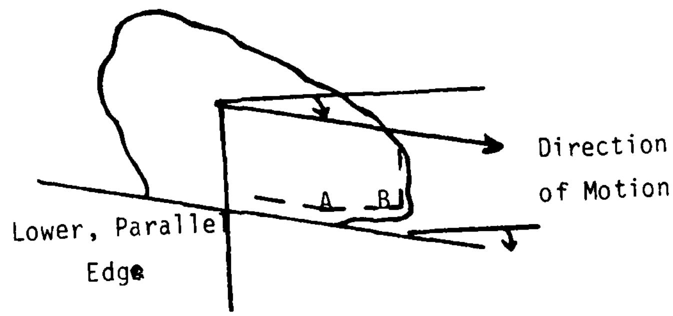


Figure 2-9. Motion-Edge Orientation Angles

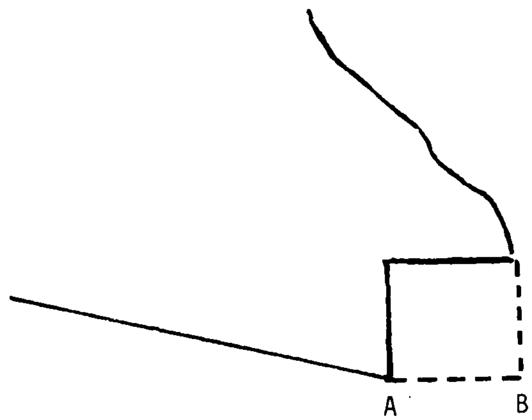


Figure 2-10. End Approximation

2.4 AN OVERVIEW OF AIMPOINT SELECTION

The purpose of this section is to describe aimpoint selection and homing from detection range to impact using the approach described in the Fourth Quarterly Report and prior sections of this report. Previously, various parts of the problem have been considered and proposed solutions discussed. Each paragraph below represents a significant range mark in the closing trajectory.

2.4.1 Detection Range

Detection range for the imaginary FLIR sensor described in the Fourth Quarterly Report was 2700 meters. Steering should be initiated at these ranges upon the detection of a target or group of targets in order to keep them within the field of-view. A particular formation of targets, the direction of motion, and hot spots in the rear portion (determined from motion) of the target will add credibility to a turn initiated at detection range and beyond classification range.

2.4.2 Classification Range

At a classification range of 1600 meters for the imaginary FLIR sensor, target aspect is computed on the classified targets. For the tank, the silhouette is 8 to 10 lines high and between 10 to 20 pixels wide. Assume that the exact width of the target is not known. The uncertainty is caused by gradient descriptions which are ramp functions, and segmentation processes which tend to use edge values based on the steepest part of the gradient curve. At 1600 meters, the target subtends about 1.5 milliradians in the vertical dimension and 3.0 milliradians in the horizontal dimension. The maximum turn rate for a 3g missile is 3.4^{0} per second or 186 milliradians per second so the missile can be easily steered to any aimpoint on the target. At classification range, the expected performance capability allows placing the aimpoint within one or two pixels of the correct location. This is satisfactory because it means that subsequent corrections will be small.

2.4.3 One-Half of the Classification Range

At one-half the classification range, or 800 meters, the target is 16 to 20 lines high and 20 to 40 pixels wide. The signal-to-noise ratio has improved by 2-3db and the target edge is tending more toward a step function than a ramp. Consequently, the exact target outline is better known. As seen from the Fourth Quarterly Report (p. 4-11), the aimpoint is fluctuating approximately $\pm$ 1 pixels about a center point. The target encompasses 3 milliradians in the vertical dimension and 6 milliradians in the horizontal dimension. The resolution is 0.3 meters per pixel. This translates into an aimpoint jitter of $\pm$ 0.3 meters and an outline uncertainty of perhaps 10% or 0.32 meters. The aimpoint is settling, and becoming more accurate based on an improved exterior edge description.The target is cued every fifth frame and tracked with a bandpass binary correlation tracker in the intervening four frames. The aimpoint is based on the binary target outline formed in the inner track window.

2.4.4 One-Quarter of the Classification Range

At one-quarter of the classification range, or 400 meters, the missile has entered the guidance settling period. Only small changes, if any, are permitted in the missile steering and the aimpoint selection process should be completed. The target has doubled again in size and in number of lines, and the target outline is better defined. The tracking has switched over from correlation tracking to edge tracking as described in the Fourth Quarterly Report. The discrepancy in handover was approximately one pixel. At this range the target is 32-40 lines high and 40-80 lines wide. Each line represents 0.15 meters and the handover from correlation to edge tracking is good to 0.15 meters. Further, as was also shown in the Fourth Quarterly Report, final tracking based on target edges for 64x64 pixel targets is also accurate to 1 pixel or 0.15 meters and was based on 25 edges. Since the guidance settling period has started the missile cannot switch to another target in the field of view; tracking is now confined to the primary target, and the remainder of the field of view can be ignored. The frame rate capability of the edge-based tracker is increased because of the smaller field of view. This does not imply that classification decision should be held off until just before 1/4 classification range.

The results of Section 2.4 and missile dynamics will be considered in the next quarter.

3.0 PRELIMINARY RESULTS

This section presents an analyses of two image processing problems alluded to in the previous section. The first, discussed in Sections 3.1 and 3.2, concerns target aspect determination. The second, described in Section 3.3, concern closein homing (>24 lines) and the registration accuracy under conditions of emerging interior features, target movement and scale changes.

3.1 ASPECT DETERMINATION FOR A MANEUVERING TARGET

The purpose of this paragraph is to consider aspect determination for a maneuvering target; i.e., the target aspect changes with time. Further, the target will have portions of the end, side, and top visible The target motion and orientation angles of the target's edges (in the image plane) are used to determine aspect. Since the data used is extensive, the section is divided into several parts. The first part describes the input data and the second describes the analysis. The third part provides an additional example

3.1.1 Input Data

The data is from the 875 line FLIR data base supplied by NV&EOL. It is found at position 5080 on video tape "#5 11/15/77", described in the First Quarterly Report. A tank, initially moving on a road, moves off onto the shoulder as shown in Figure 3-1. Another vehicle, a jeep, is moving ahead of the tank and stays on the road The tank passes the jeep in the last part of the sequence. The helicopter appears to be hovering with an initial declination angle (between horizontal and the tank) of perhaps 30^0 As the tank approaches the hovering helicopter the declination angle increases to an estimated 80^0. The portion of interest is shown in the ellipse formed by the dashed lines in Figure 3-1.

The Intelligent Tracker is cueing every fifth frame and tracking targets on the intervening four images with a binary correlation tracker. This is part of the system concept for the Intelligent Tracker described in Section 1.0. The data used for the analysis starts at the 778th frame from the beginning of the sequence; every fourth frame was digitized because of the long length of the sequence. Further, a single field was digitized from each frame to reduce the amount of data handled. Aspect determination is analyzed for images 778, 782, 786 ... . 838, where images 778, 798, 818, and 838 are cued and the other images are tracked.

Figure 3-2a shows the segmentation for the tank in image 778; the outline of the target is constructed in Figure 3-2b. The outline was hand drawn in this case but there are algorithms available for linking edges, one of which was described in the First Quarterly Report. Superlink[1] is another edge linking algorithm based on the segmentation format presented by Superslice, the segmentation algorithm used extensively in the present work. The track windows are shown in Figure 3-3, 4, ... 7. Figure 3-3 is the track reference window for image 778, and Figures 3-4, 5, 6, 7 are the track windows for images 782, 786, ... 794. The calculations under each

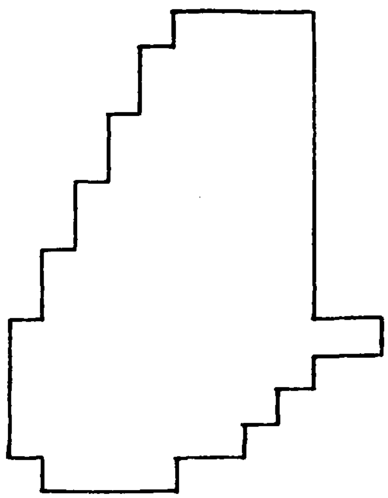


Figure 3-2a. Segmented Tank

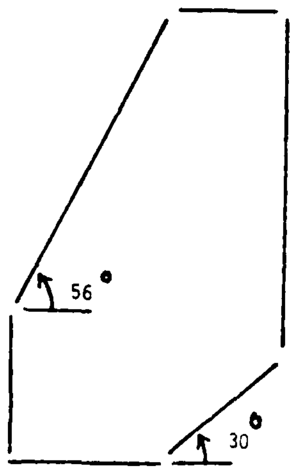
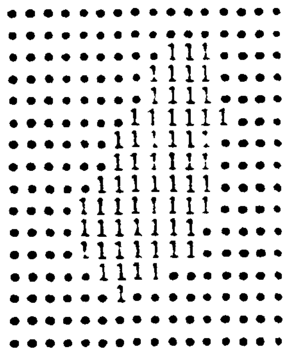
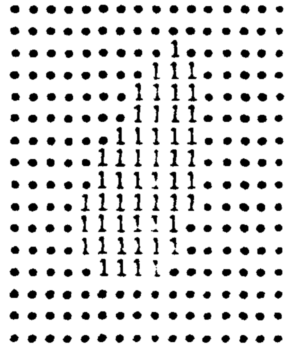


Figure 3-2b. Tank Outline



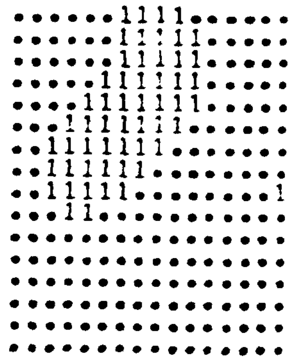
```
XBAR   =    9.111
YBAR   =    8.587
SIG2X  =    4.289
SIG2Y  =    8.465
SIGXY  =   -3.113
THETA  =  -28.276
SIG2XP =    2.523
SIG2YP =   10.125
SZX    =    5.617
SZY    =   11.223
```

Fig. 3-3. Image 778



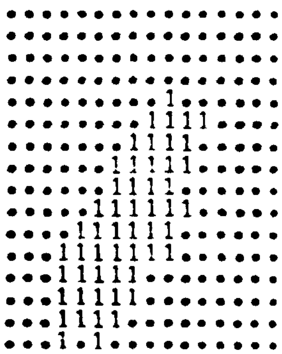
```
DELX   =       -1     DELY =
DIRFC  =      135
XBAR   =    8.442
YBAR   =    8.769
SIG2X  =    3.293
SIG2Y  =    7.372
SIGXY  =   -2.226
THETA  =  -22.443
SIG2XP =    2.139
SIG2YP =    8.523
SZX    =    5.233
SZY    =    9.982
```

Fig. 3-4. Image 782



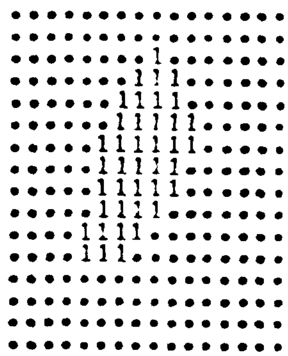
```
1 DELX   =       -2     DELY  = -3
  DIRFC  =      237
  XBAR   =    7.364
  YBAR   =    5.473
  SIG2X  =    6.377
  SIG2Y  =    6.504
  SIGXY  =   -2.454
  THETA  =  -44.384
  SIG2XP =    3.486
  SIG2YP =    9.395
  SZX    =    6.468
  SZY    =   10.318
```

Fig. 3-5. Image 786



```
DELX   =       -1     DELY =
DIRFC  =      117
XBAR   =    7.560
YBAR   =   10.755
SIG2X  =    4.639
SIG2Y  =    8.497
SIGXY  =   -4.327
THETA  =  -33.466
SIG2XP =    1.553
SIG2YP =   11.473
SZX    =    4.334
SZY    =   11.734
```

Figure 3-6. Image 790



```
DELX   =       -1     DELY =      -1
DIRFC  =      226
XBAR   =    8.330
YBAR   =    7.830
SIG2X  =    2.533
SIG2Y  =    5.963
SIGXY  =   -2.353
THETA  =  -24.919
SIG2XP =    1.548
SIG2YP =    6.912
SZX    =    4.339
SZY    =    9.138
```

Figure 3-7. Image 792

track frame were discussed in detail in the Fourth Quarterly Report and are briefly reviewed here. The quantities XBAR and YBAR are the coordinates of the target centroid as measured from the upper lefthand corner of the track window with the convention $\downarrow_{+y} \rightarrow^{+x}$. The quantities SIG2X, SIG2Y, and SIGXY are the second moments (statistical variances) of the target, and $\sqrt{\text{SIG2X}}$ and $\sqrt{\text{SIG2Y}}$, standard deviations, are a rough measure of the target X and Y dimen- sions in the image plane coordinates. Theta is the orientation angle of the prin- cipal axis as measured around the centroid. In this case, theta measures the ori- entation of the principal X axis from the X axis in the image plane as seen in Figure 3-8. The quantities SZX and SZY measure the X and Y target dimensions in the directions of the principal axes. The quantities DELX, DELY, and DIREC measure the movement of the track window from frame to frame in the coordinate system of the image plane.

As an aside, a technical point regarding data presentation must be made. Note that the target images in the track window are slightly smaller than those of the segmentation image. The target was close to the bottom of the digitized window, so a small track window had to be used. Hence, the Superslice threshold was increased one level to present a smaller target and facilitate a smaller window. The effect on the results was not significant as the principal axes orientation angles remain- ed approximately the same. This problem is artificial. In obtaining this test data, the sensor operator did not center the target as the cuer would have done.

To complete the discussion of the initial data, note Figure 3-9, which is the 128 x 128 pixel window grabbed from image 778. Figure 3-9 shows the target for image 798 roughly drawn in to show the amount of target movement from image 778 to image 798. Consider the method used to obtain the amount of target movement from image 778 to 798.

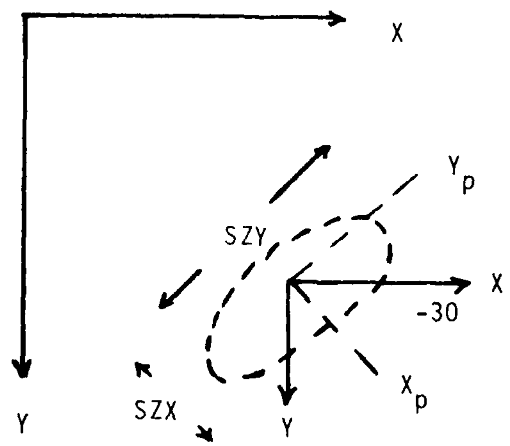


Figure 3-8. Principal Axis Orientation

3.1.1.1 Determination of Target Movement

This description is included under the Input Data paragraph because change detection has been discussed in detail in the Second and Third Quarterly Reports. However a variation of those techniques is applied here which appears to be an improvement. Figure 3-10 shows the segmented target and portions of the background for image 778. Figure 3-11 is the counterpart for image 798 Note that the numbers under the image provide statistics for each seg- mented object shown in each image. The BLOB statistic describes which object (blob) is being described by the particular row of statistics. AVG is the average gray level of the object. The quantities LFTX (Left x), TOPY (Top y), RGTX (Right x), and BOTY (Bottom y) describe the X and Y coordinates at the extremes of each object. This was

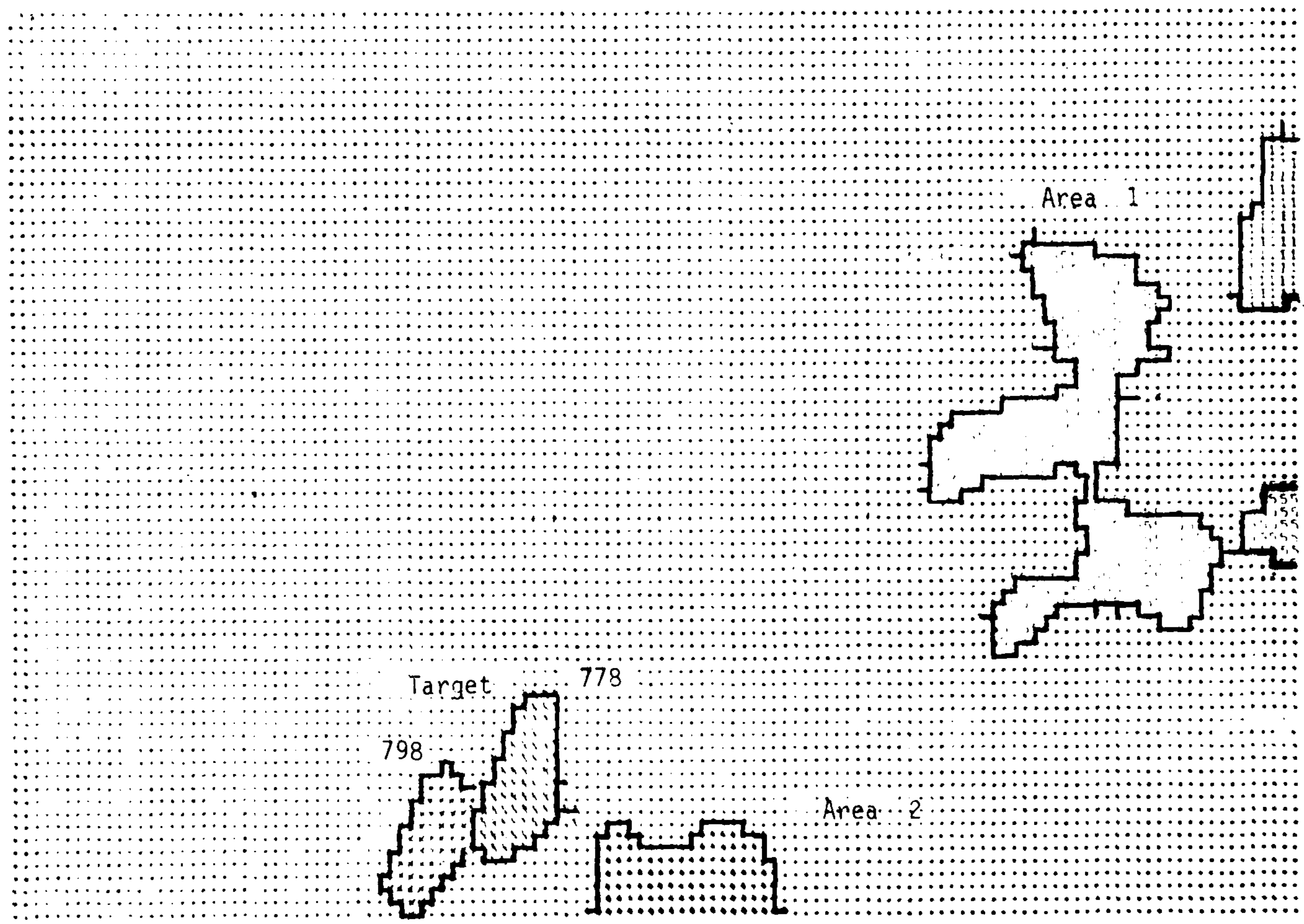
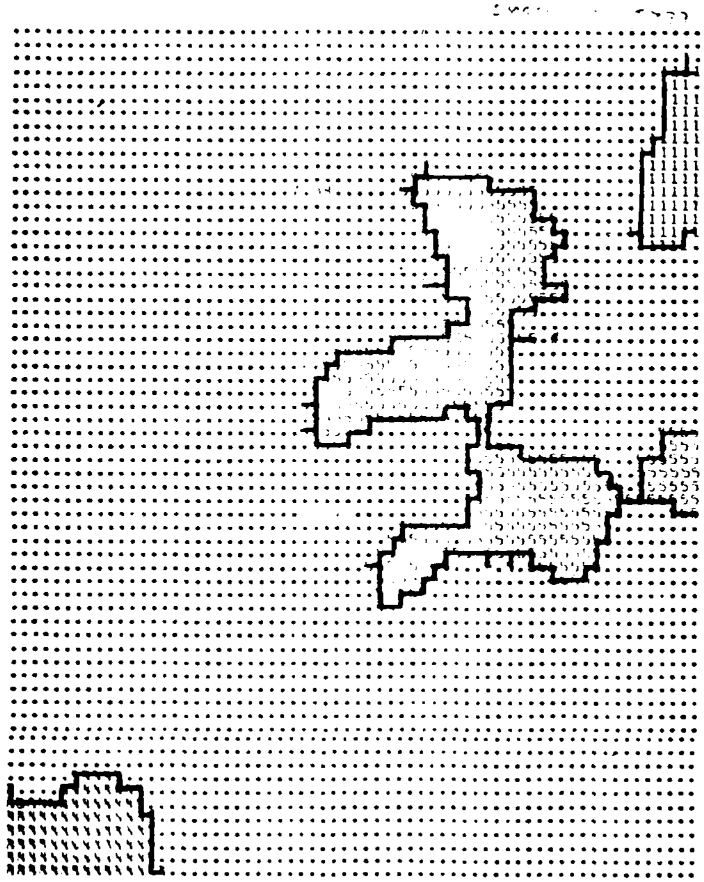


Figure 3-9, 12. Target for Image 798 Superimposed on Image 778.

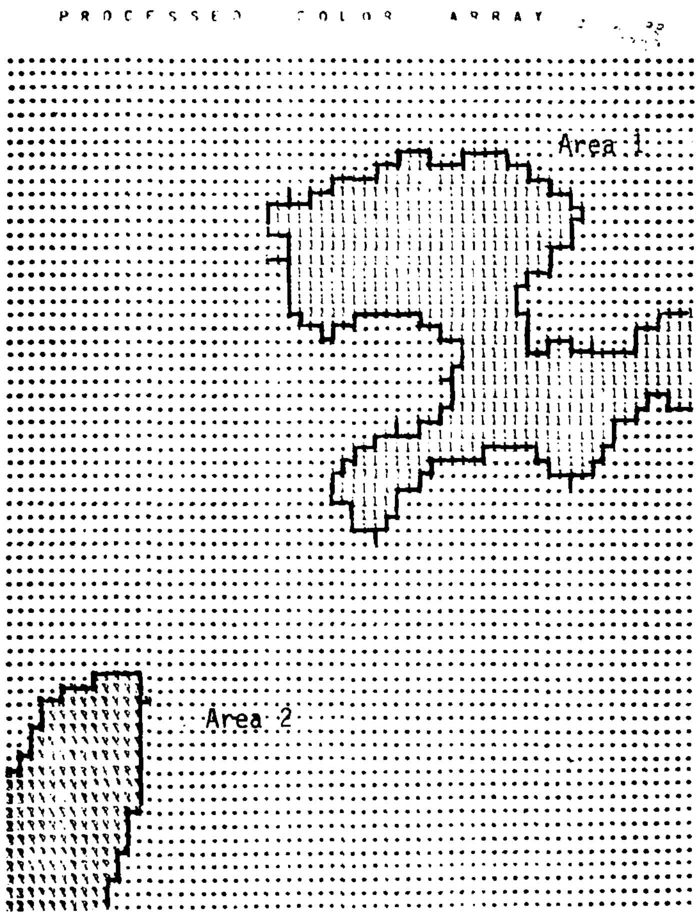
P R O C E S S E D   C O L O R   A R R A Y



B L O B   S T A T I S T I C S

BLOB		AVG	LFTX	TOPY	[illegible]	[illegible]	[illegible]	YBAR	DELX	DELY	SIZE	PERPTS
1.	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	7	15	73	[illegible]
2.	5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	34	491	214
3.	7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1	1	1	1
4.	8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2	1	2	2
5.	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1	1	1	1
6.	8	[illegible]	[illegible]	[illegible]	77	[illegible]	[illegible]	[illegible]	15	9	[illegible]	42

Figure 3-10. Segmentation Results for Image 778



BLOB STATISTICS

									CELX		SIZE	PERPTS
1.	1	[illegible]	47	7	[illegible]	[illegible]	[illegible]	[illegible]	40	[illegible]	605	[illegible]
2.	2	[illegible]	63	[illegible]	[illegible]	[illegible]	[illegible]	117.96	14	[illegible]	138	[illegible]

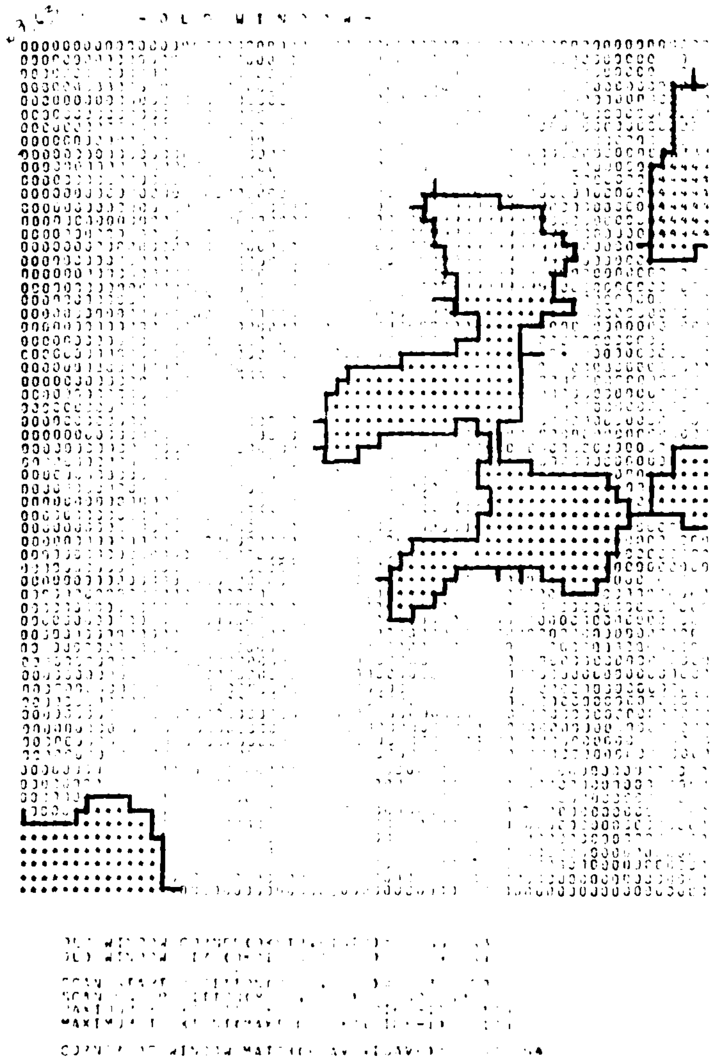
Figure 3-11. Segmentation Results for Image 798

developed to provide perspective information for target signature prediction and is described in detail in the Second Quarterly Report, page 4-9. The data tends to indicate the distance between the target and obscuration and whether it will pass in front of or behind the obscuration. For example, BOTY1>BOTY2 means that target 1 is below target 2 in the image. When they cross, target 1 will be in front. The quantities XBAR and YBAR are the X Y coordinates of the objects' centroid. DELX and DELY are the X and Y dimensions of the object and SIZE indicates target area in pixels. PERPTS is the number of perimeter points (object pixels having at least one neighboring pixel at the zero gray level) in the object. Suppose the objects are renamed T for target, Area 2 for that portion of background in the lower part of the window, and Area 1 for that portion of the background in the right side of the image and above the target. The new nomenclature is shown in Figure 3-12, page 3-7. Figure 3-13 shows the X,Y coordinates (XBAR, YBAR) for the centroids of the target, AREA 1, and AREA 2 for both images.

	Target		Area 1		Area 2	
IMAGE	XBAR	YBAR	XBAR	YBAR	XBAR	YBAR
778	52	116	107	91	69	123
798	43	110	105	82	69	117
x	-09		-02		0	
y		-06		-09		-06

Figure 3-13. Object characteristics for Images 778 and 798

The amount of change (Δx, Δy) in XBAR and YBAR for each object is shown in Figure 3-13. The target moved nine pixels to the left and six pixels up; AREA 1 moved two pixels to the left and nine pixels up; and AREA 2 remained stationary in x and moved six pixels up. The changes indicate that the objects, as a collection, moved to the left and up in the coordinate system of the image plane which is positive to the right in x and positive down in y, $\begin{smallmatrix}\rightarrow x+\\ \downarrow y+\end{smallmatrix}$, as measured from the upper left hand corner of the image. The amount of change indicates the range over which the maximum correlation for binary change detection will be obtained. For example, the maximum correlation in x should occur in a neighborhood about the interval -09 to 0; the maximum correlation in y should occur in a neighborhood about the inteval -06 to -09 Hence, there is no need to search an interval, e.g. of -32 to +32 pixels in x, and -32 to +32 pixels in y. If a neighborhood of $\pm$ 5 pixels were chosen about the changes, the range of possible correlations would be -14 to +5 in x and -01 to -14 in y. This reduces the number of correlation computations for binary change detection from 4096 (64x64) to 285 (19x15). Second, a window encompassing AREA 1 and AREA 2 and not the target is used for change detection to obtain scene movement caused by sensor motion. Figure 3-14 shows the computer printout from the binary change detection for a 64x64 window containing AREAS 1 and 2. The windows were matched best at -1, -9 as shown in Figure 3-14. This means that AREAS 1 and 2 for image 798 were one pixel to the left and nine pixels above (-1,-9) the locations of AREAS 1 and 2 for image 778. The target position for image 798 was nine



OLD WINDOW CORNER (OKSTR,OISTR)= 63 63

CORNER OF WINDOW MATCH (SAVE,ISAVE)= 62 54

Figure 3-14. Binary Change Detection Printout pixels to the left and six above the target location for image 778. But this motion included target motion plus sensor motion. The former (-1,-9) displacement is for sensor motion only. Hence, sub- tracting the two, (-1,-9) (-9,-6), gives (-8,+3) as target displacement alone This is to the left and down.

In conclusion, the magnitude of the binary change detection computation used here was limited in window size to a 64x64 (encompassing AREA 1 and AREA 2) and in correlation range by using information already developed about the images. Further, the x, y information developed for AREA 1 and AREA 2 is a rough early indication of the change detection results.

3.1.2 Analysis

The input data for target aspect determination has been obtained. The target has been segmented, and an edge outline has been completed. The target dimensions about the principal axes, the orientation angles of the principal axes and target edge outlines in the image plane, and direction of motion have been obtained. Note that the only supplemental algorithm necessary is the edge orientation algorithm, one form of which we described in the Fourth Quarterly Report and which is used in close-in homing. Hence, we are not introducing any new algorithms for target aspect determination but rather using only available information. Figure 3-15 shows the orientation angles in the image plane coordinates (x_i, y_i), for the case at hand. The orientation angle of the principle x axis, xp, is -28^o, -23^o, -44^o, -33^o, and -25^o from the track windows shown in Figures 3-3 through 3-7. This is represented at $\theta = 30^o$ in Figure 3-15. Further, the total length of the target along the principal x and y axes is 5 and 10 pixels, respectively, which falls within the range of aspect ratios for tanks. The results of the change detection record show that the target movement from image 778 to image 798 is approximately -8 and +3 in x_i and y_i.

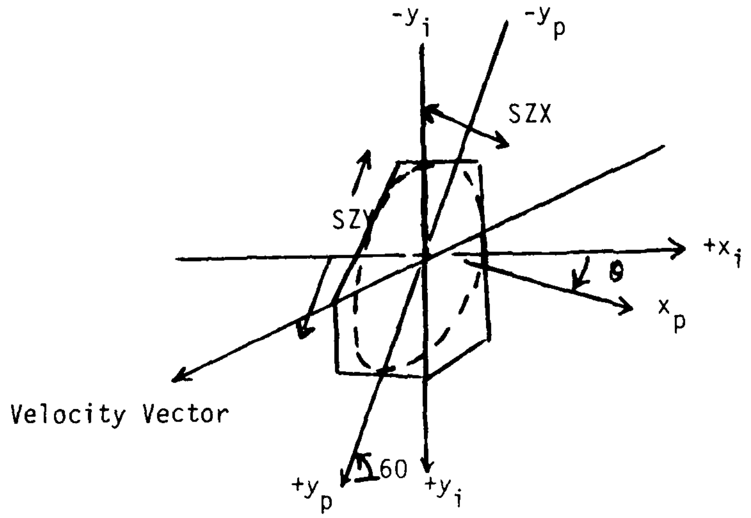
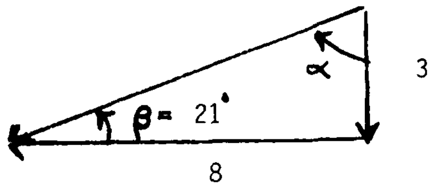


Figure 3-15. Orientation Angles in the Image Plane

Figure 3-16 shows the movement. Hence, the orientation angle of the target velocity vector with respect to the $+y_i$ axis is -69^0. Compare the orientation angles of the velocity vector and principal axes orientation with those of the target edges shown in Figure 3-2b and redrawn in Figure 3-15. To put the angles in a common coordinate system, we measure the angles from the horizontal and in a counterclockwise direction. The lower slanted edge of the target has an orientation of approximately 30^0 and the target velocity vector has an orientation of 21^0 in the same system. The upper slanted target edge has an orientation of 56^0 and the principal y axis has an orientation of 60^0 in the same system. Hence, a number of conclusions can be drawn about the target, as follows.

The sides of the tank are generally oriented to align with the direction of motion of the tank in straight motion. Hence, that edge most closely aligned with the velocity vector (edge 3) probably represents the side of a tank. The lower part of the tank is the front of the tank because it is in the direction of motion; the upper part of the tank is directed



$\alpha = \tan^{-1}(8/3) = 69°$

Figure 3-16. Target Movement Direction

oppositely. The relative orientations of edges 1, 2, and 3 plus the suppositions that edge 3 is the side of the tank and the lower part of the tank is the front, allows the construction of the dotted lines as shown in Figure 3-17 which complete the orientation for the oblique view presented. The orientation of edge 4 is not aligned with that of edge 3 because it must account for the presence of the turret. Implementation will be considered in the Final Report.

Having presented data and analyzed target aspect for images 778 through 798, consider the techniques just developed as applied to 798 versus 818.

3.1.3 Images 798 versus 818

The scenario just described is now analyzed over the next set of tracked images: 802, 806, 810, 814 and the cued image 818. This is a particularly interesting set because the target is turning and changing aspect, in the presence of clutter or a potential obscuration. A conflict can arise between obtaining a complete target outline for aimpoint purposes with a higher risk of track loss, and obtaining a less descriptive target outline and a lower risk of track loss.

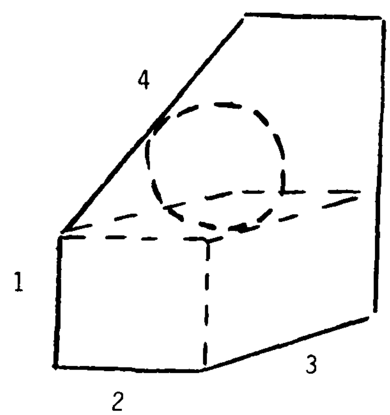


Figure 3-17. Oblique View of Tank

The purpose of this section is to examine the problem of target aspect determination for a maneuvering target in the presence of potential obscurations. First, the problem is described in the context of the binary bandpass tracker. Second, images 802, 806, 810, and 814 are presented and the problem described in terms of the data. Third, the proposed approach to aspect determination is described. This approach is the same as that described in the previous section. Fourth, the proposed approach is applied to the data and the results described. Finally, an amendment to the intelligent tracker system concept is described which ties together two parts of the concept whose relationship was brought out by this test data. The relationship has to do with the handover between the cuer and tracker.

Problem description - In the previous intelligent tracker work, emphasis has been placed on track continuity by using a cuer coupled with a bandpass binary correlation tracker. The tracker consists of an interior and exterior window as shown in Figure 3-18. The interior window is placed over the cued target such that 90 percent of the target is within the interior window. An exterior window is selected and centered on the interior window, which is double the height and width of the interior window. A histogram of the exterior window represents the background. A discriminant function is constructed, (e.g. Neyman-Pearson), which maximizes the metric (distance) between the two histograms. Essentially this has the effect of eliminating gray levels which are common and have substantial representation in the background histogram. For pixels in the exterior window at the same level as the target, these procedures eliminate those levels from the target. The result is a less than accurate description of the target, a less accurate estimate of target aspect, and an erroneously placed aimpoint.

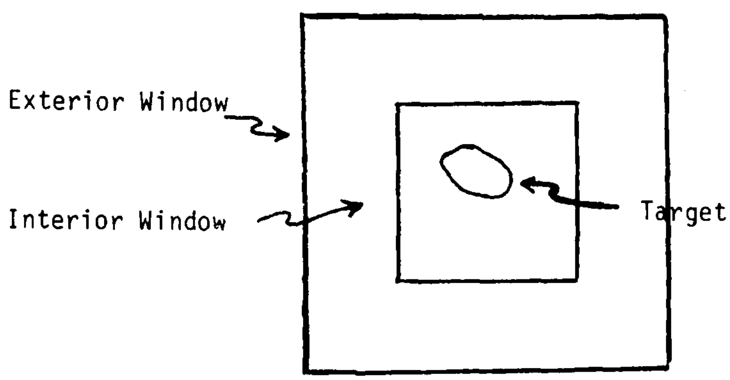


Figure 3-18. Track Window Arrangement

Data presentation - The cued target for frame 798 is shown in Figure 3-19. The target was thresholded at 39 for a maximum edge/perimeter match. The corresponding binary reference window is shown in Figure 3-20 and has been thresholded by the tracker at 40. The reason for this is shown in Figure 3-21, which is a composite of the interior and exterior windows. Figure 3-21 shows the target thresholded at 39 and 40; it also shows the presence of background at the 39 level in the lower right hand corner of the exterior window. This is the reason for the new thresholding by the tracker. The track windows for images 802, 806, 810, and 814 are shown in Figures 3-22, 23, 24, and 25. Recall, the target aspect determination approach described in 3.1.2. Edge orientation, motion, and principal axis orientation are used primarily to determine aspect. Figures 3-26, 27, 28, and 29 show the target

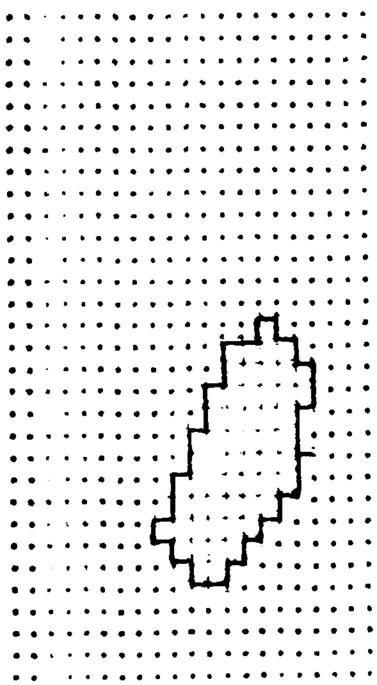


Figure 3-19. Cued Target Image 798

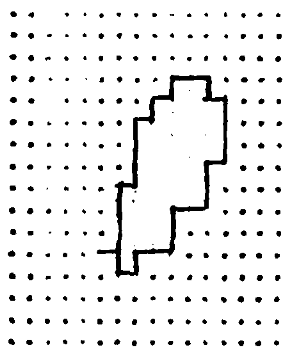


Figure 3-20. Binary Reference Window Image 798

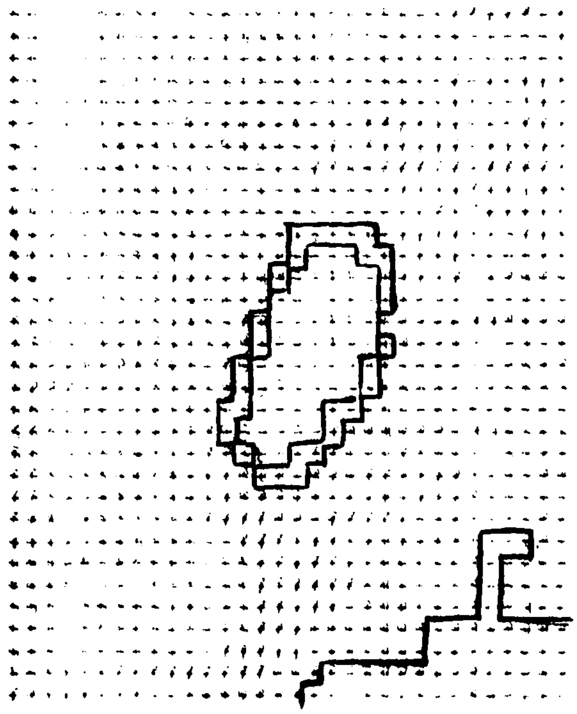


Figure 3-21. Composite Windows, Image 798

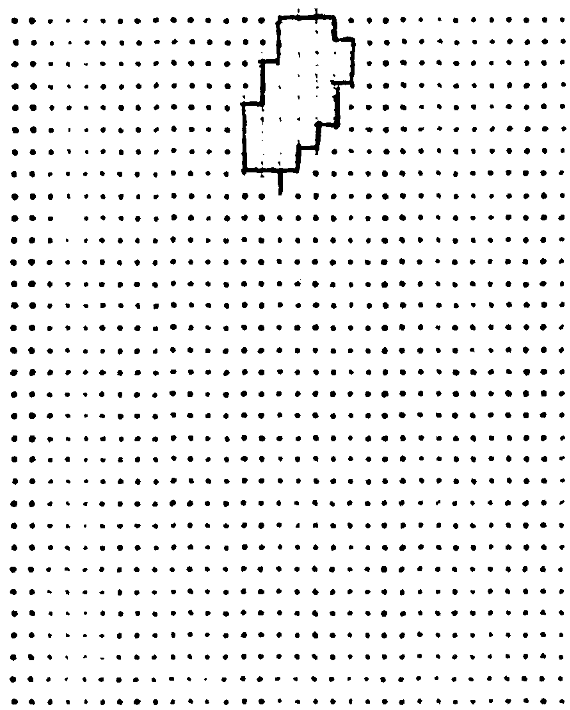


Figure 3-22. Track Window, Image 802

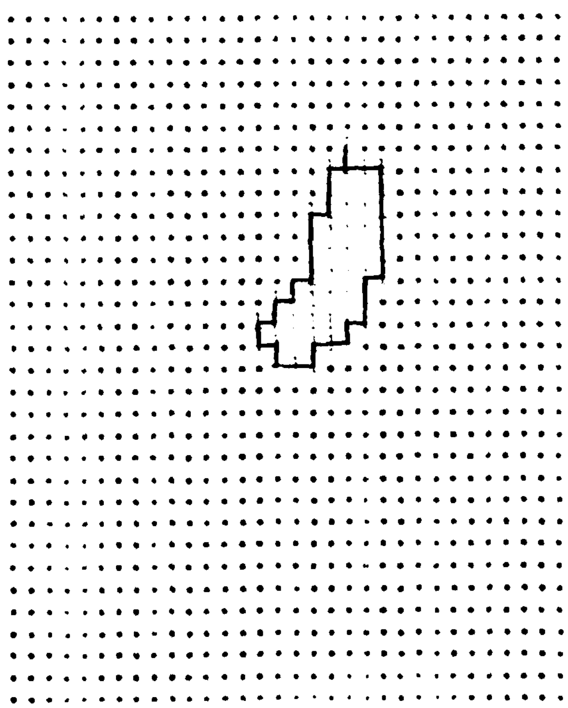


Figure 3-23. Track Window, Image 806

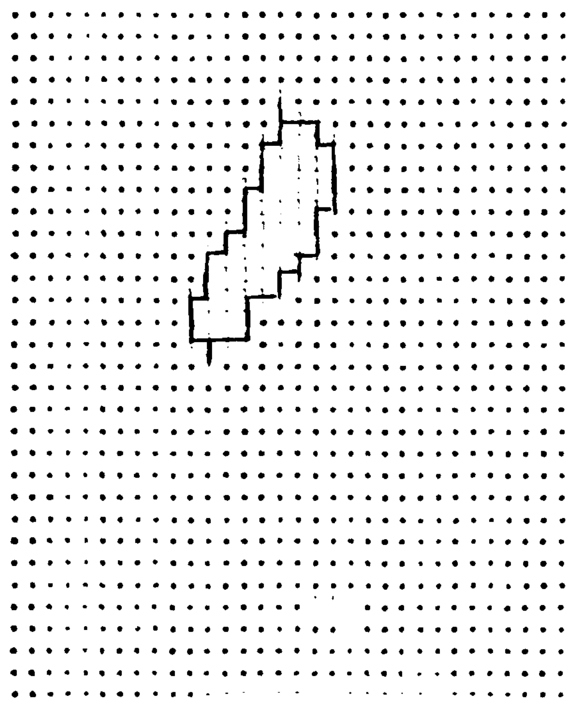


Figure 3-24. Track Window, Image 810

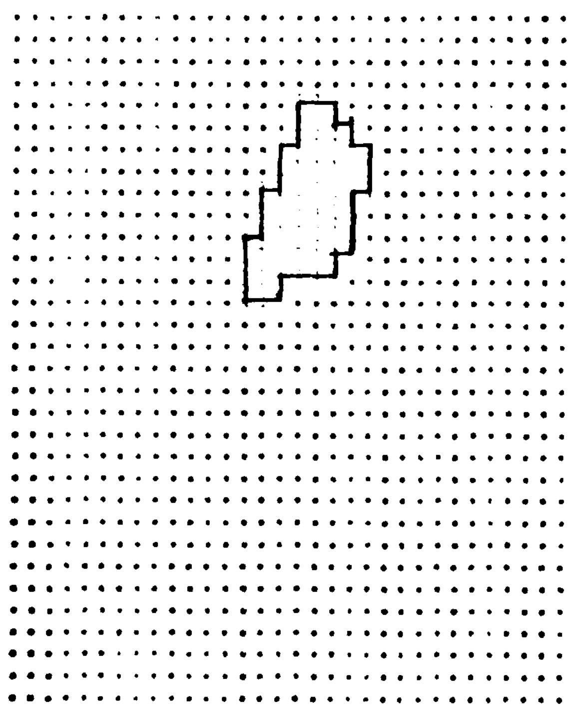


Figure 3-25. Track Window, Image 814

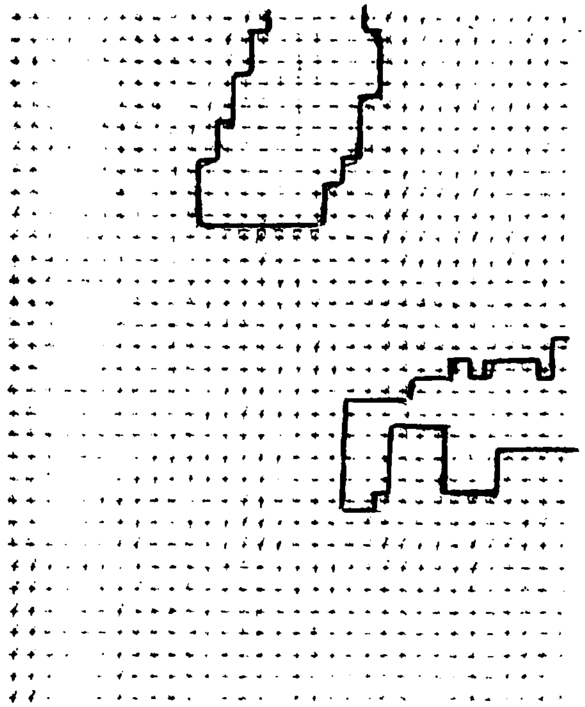


Figure 3-26. Target, Image 802

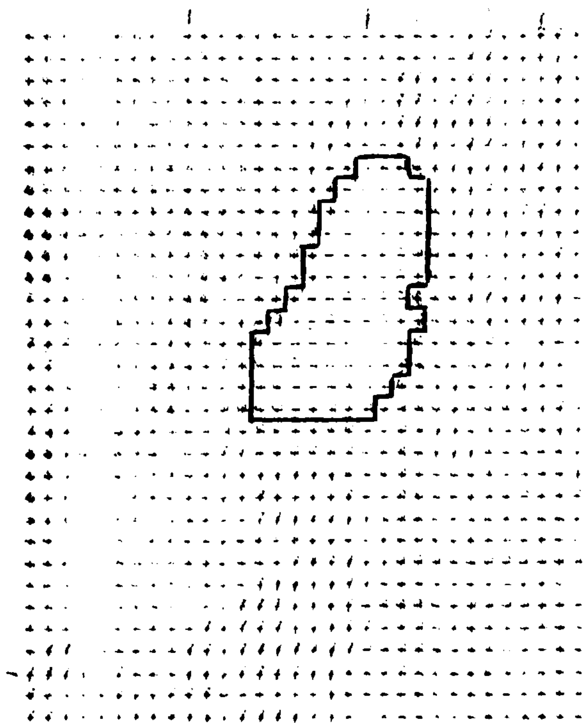


Figure 3-27. Target, Image 806

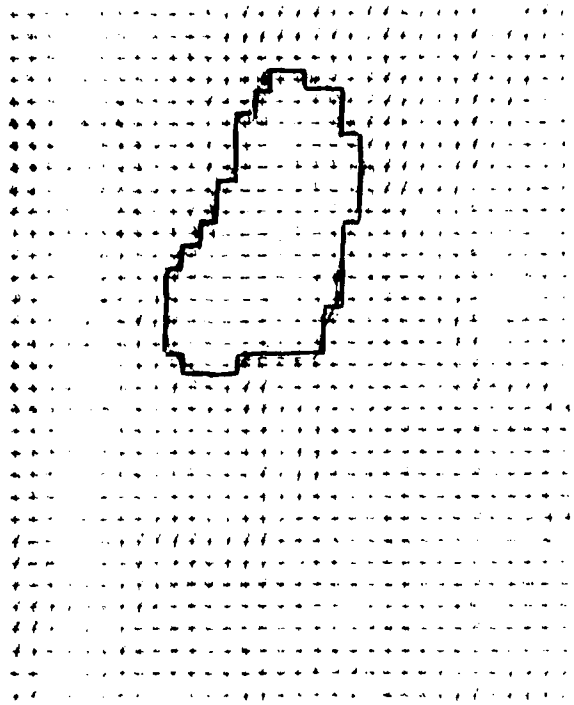


Figure 3-28. Target, Image 810

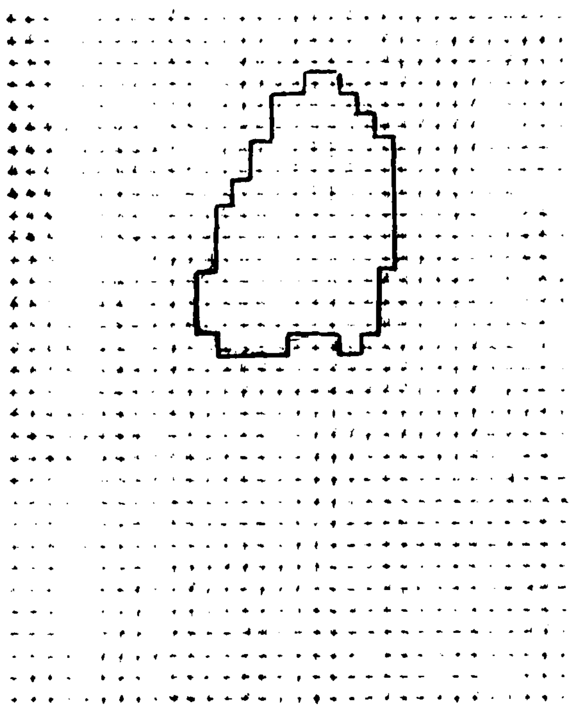


Figure 3-29. Target 814

Target Histogram

0.	0	9.	0	18.	0	27.	0	36.	0
1.	0	10.	0	19.	0	28.	0	37.	0
2.	0	11.	0	20.	0	29.	0	38.	0
3.	0	12.	0	21.	0	30.	0	39.	28
4.	0	13.	0	22.	0	31.	0	40.	36
5.	0	14.	0	23.	0	32.	0	41.	13
6.	0	15.	0	24.	0	33.	0	42.	0
7.	0	16.	0	25.	0	34.	0	43.	0
8.	0	17.	0	26.	0	35.	0	44.	0

Sector 1 Histogram

0.	0	9.	0	18.	0	27.	0	36.	16
1.	0	10.	0	19.	0	28.	0	37.	11
2.	0	11.	0	20.	0	29.	0	38.	7
3.	0	12.	0	21.	0	30.	0	39.	0
4.	0	13.	0	22.	0	31.	0	40.	0
5.	0	14.	0	23.	0	32.	20	41.	0
6.	0	15.	0	24.	0	33.	24	42.	0
7.	0	16.	0	25.	0	34.	19	43.	0
8.	0	17.	0	26.	0	35.	17	44.	0

Figure 3-30. Histograms

at a threshold of 40 for the same set of images. There is a marked difference between the two sets of images. Note in particular, the orientation angle and extent of the bottom horizontal edge. The first set of images indicates that the target is roughly going in the same direction and orientation, while the second set indicates that the vehicle is turning and changing aspect.

Proposed approach - To solve the above problem, we make use of the tools already constructed for the intelligent tracker and described in the Third Quarterly Report. The direction of motion of the target has been determined as described in the previous section. Based on this motion, histograms are constructed ahead of the target at approximately the same height (line) as the target. The histograms are shown in Figure 3-30 for image 798, and are

used to predict target obscurations. Note that these histograms do not predict obscurations at threshold 39 in the direction of motion. Hence, the correlation tracker can be overridden and the cued threshold used for the tracker. This relationship between the target signature histograms and the exterior track window was brought into focus with this set of data.

Target aspect determination - The target aspect determination techniques described in the previous sections are now applied to the images of Figure 3-26 through 3-29 (the images segmented by the cuer). The bottom edge has lengthened substantially, and the 30^{0} edge associated with the motion has become a near horizontal edge. The motion between images 798 and 818 shown in Figures 3-31,36 is now discussed. A portion of the segmented background for image 798 is shown in Figure 3-32 and the same section is shown in Figure 3-33 for image 818. A print out of the matched registered binary windows and the registration results are shown in Figure 3-34. The changes between the two windows are shown in Figure 3-35.

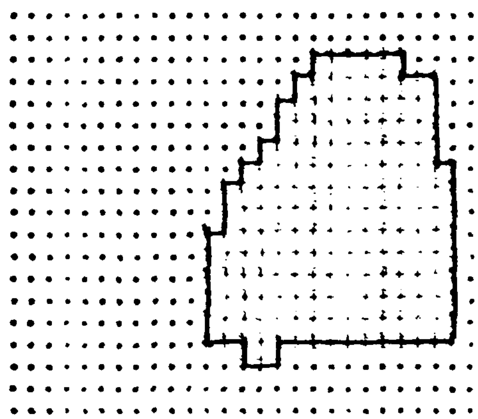
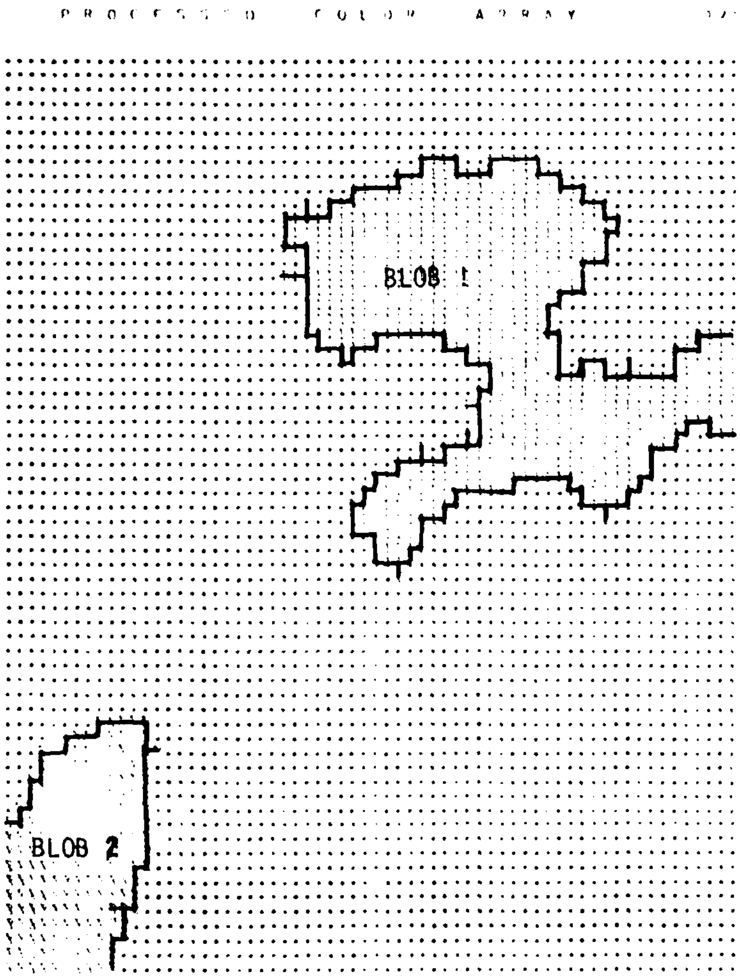


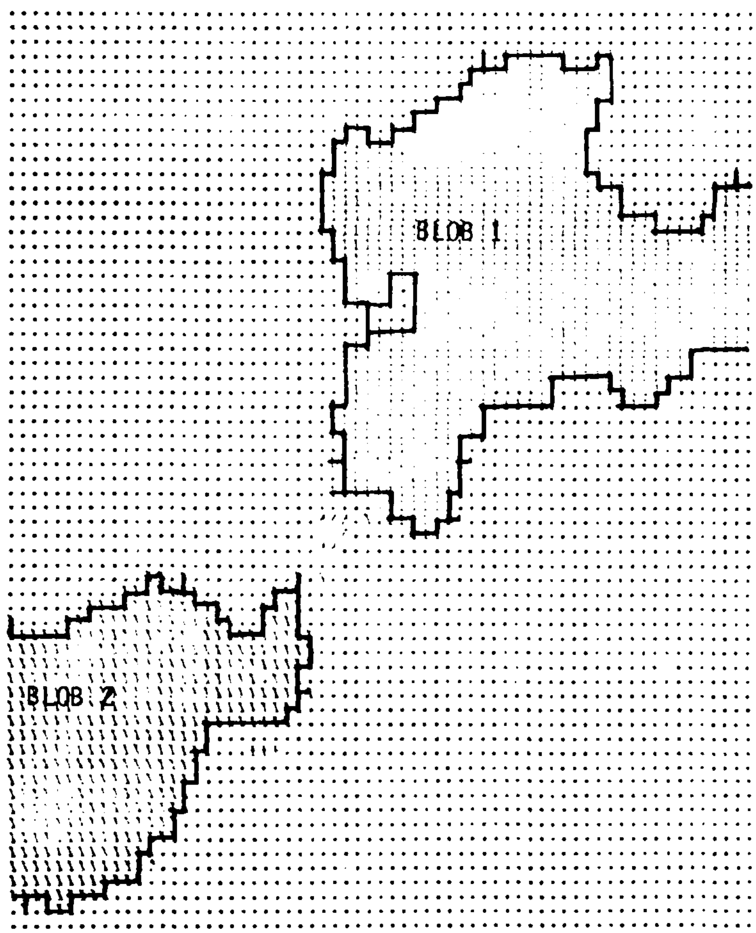
Figure 3-31. Image 818, Target



	BLOB	AVG	LFTX	TOPY	RGTX	BOTY	XBAR	YBAR
1.	1	39.57	87	70	126	99	105.48	82.57
2.	2	40.07	63	109	76	126	68.18	117.86

	BLOB	DELX	DELY	SIZE	PERPTS
1.	1	40	30	603	198
2.	2	14	18	198	59

Figure 3-32. Image 798 Background

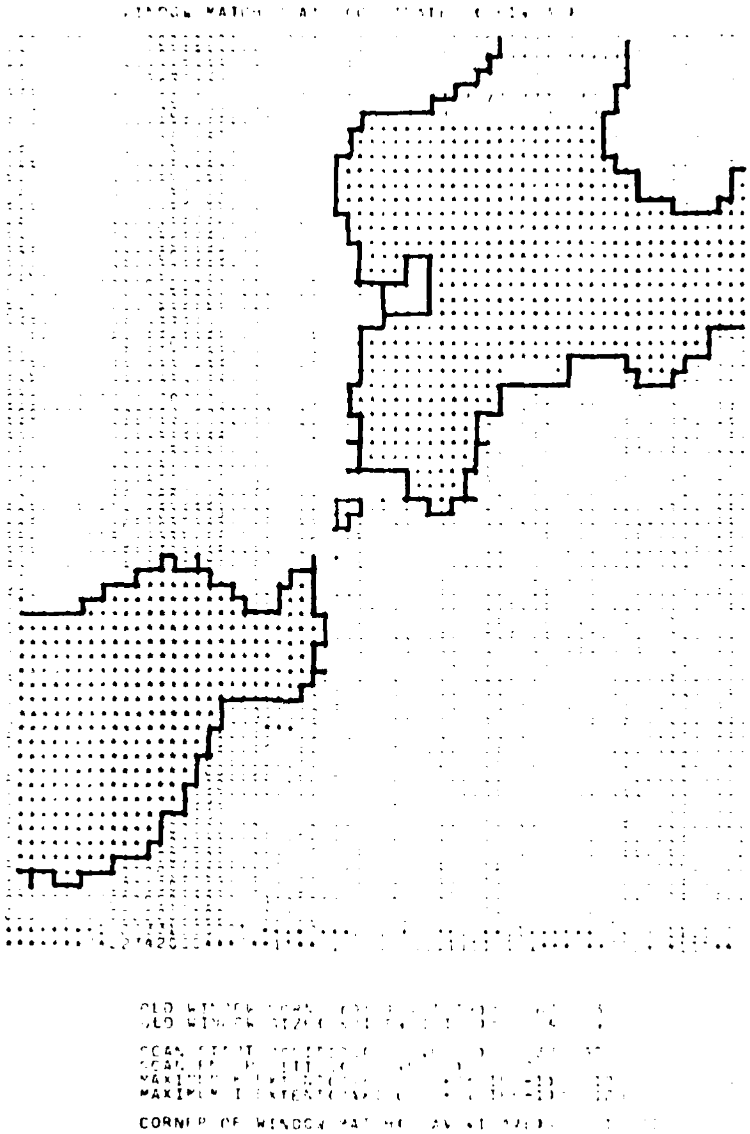


	BLOB	AVG	LFTX	TOPY	RGTX	BOTY	XBAR	YBAR
1.	1	38.70	89	30	126	63	105.27	45.16
2.	2	39.45	62	66	88	89	72.50	76.37

	BLOB	DELX	DELY	SIZE	PERPTS
1.	1	38	34	777	191
2.	2	27	24	417	117

Figure 3-33. Image 818

Figure 3-34 shows that the movement between images is (-2, -33), i.e. image 818 moves two pixel to the left and 33 up from image 798. The target movement is shown in Figure 3-34 and includes sensor movement. It is (-7, -35), hence, the net target movement is (-5, -2) and the bulk of the movement corresponds to a changed target aspect.



OLD WINDOW CORNER (OKSTR,OISTR)= 63 63

CORNER OF WINDOW MATCH(KSAVE,ISAVE)= 61 30

Figure 3-34. Matched Windows

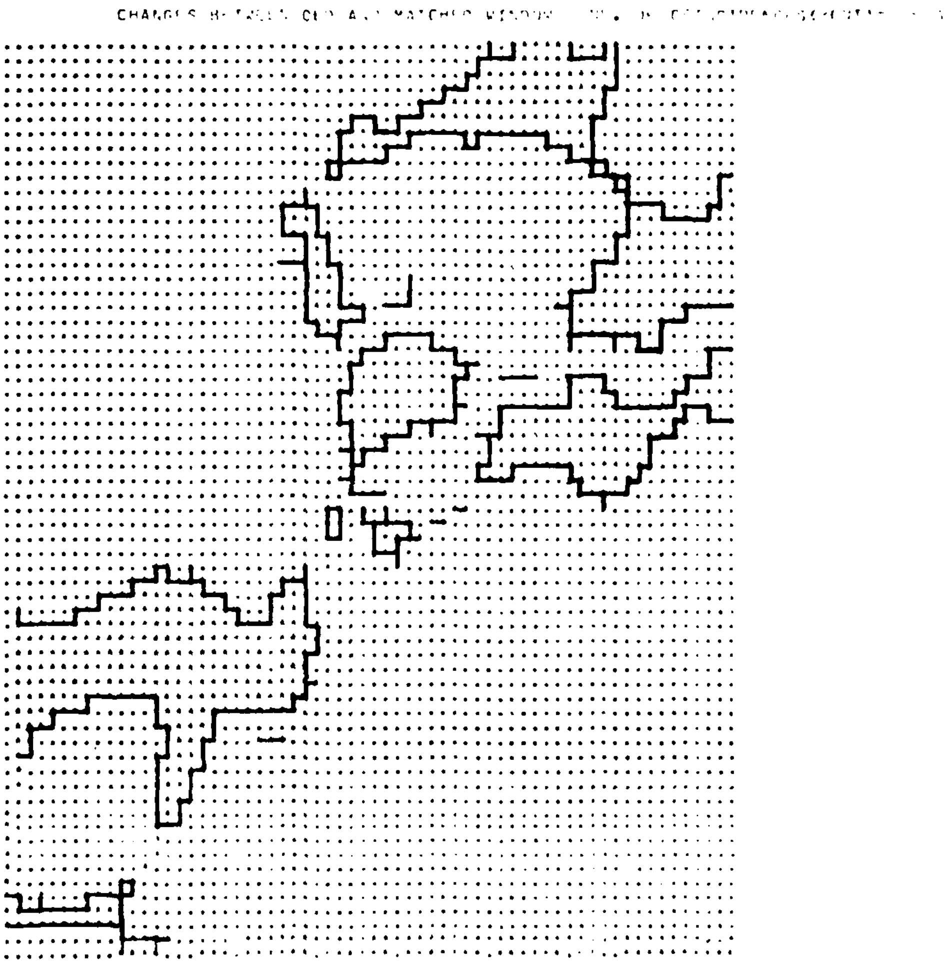


Figure 3-35. Changes Between Images

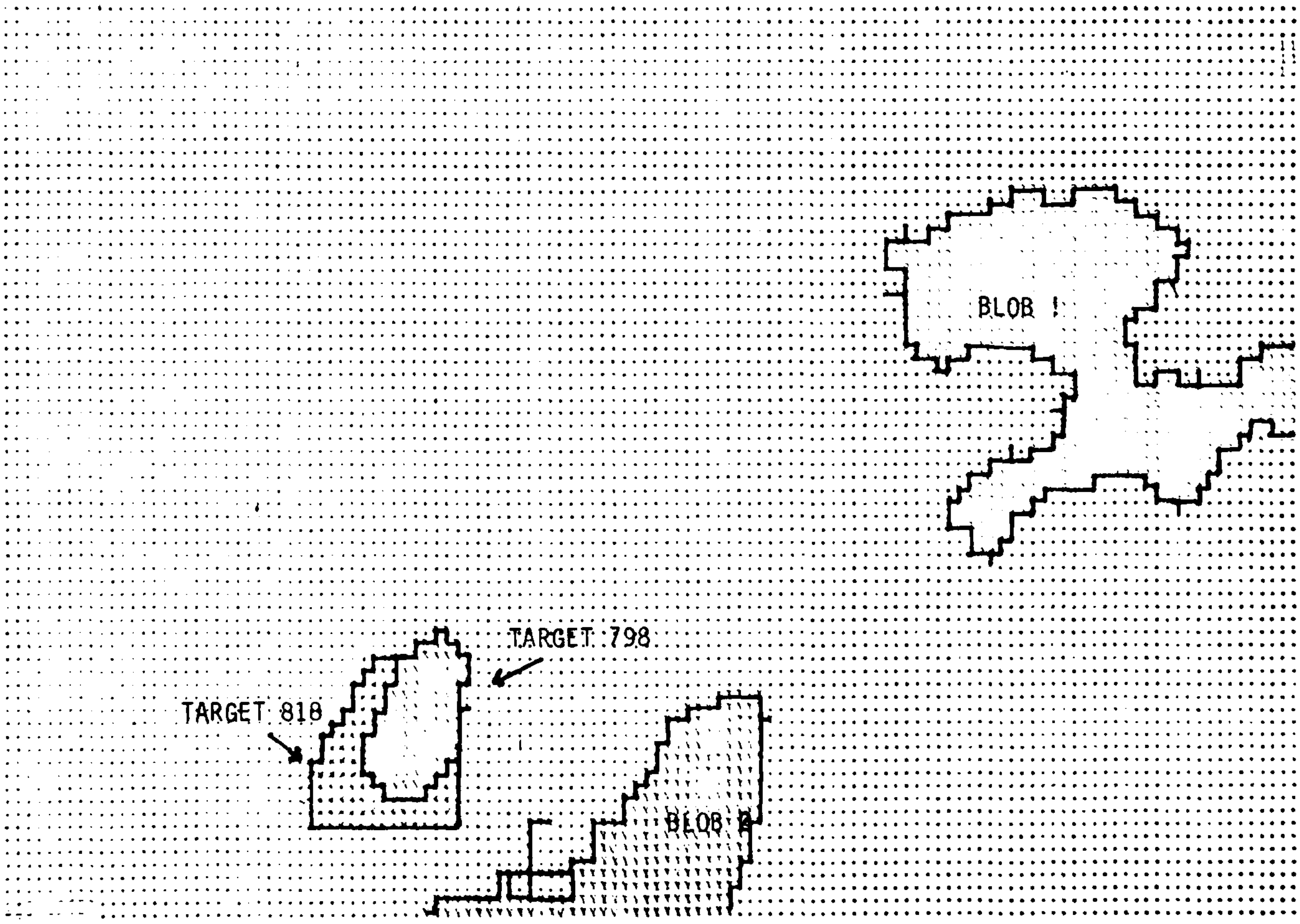


Figure 3-36. Target Movement

3.2 TRACKING STABILITY AND INTERIOR FEATURES

In the Fourth Quarterly Report, the intelligent tracker homing concept was described. It consisted of correlation tracking on the target exterior at long range and a transition to edge tracking for close-in homing. The transition took place at roughly 500 meters from impact with the target 30 lines high. The tracking accuracy was found to be approximately one pixel with a transition (hand over) accuracy of one pixel, also. The tracking in both regimes is based on the exterior target shape. In this paragraph, the analysis is taken a step further, and tracking accuracy is examined for the case of close-in homing and emerging interior features. The emerging interior features will change the target signature. Edges will be added to the interior. An important consideration is whether these added edges cause confusion or by gradually emerging reinforce the exterior edges in the registration process. If the latter is true, then the exterior edges initially overwhelm the emerging interior edges. Then as the interior edges become repeatable from frame-to-frame, they will contribute positively to the overall registration process.

Another issue in the analysis is the tracking accuracy under a scale change from frame to frame. Consider the geometry of the situation. The missile is at a range of 500 meters from the target which is 30 lines high. It is closing at a velocity of 500 meters per second. The target will be 60 lines high at 250 meters from impact, which will occur one-half second and 15 frames later. In this interval the target height is growing at 2 pixels per frame (approximately a 10% change in target area per frame depending on target aspect); closer in, the target will grow faster. This is the scale change from frame to frame.

Additionally, there is lateral movement from frame to frame caused by the motion of the target and missile. Assume the target is moving perpendicularly to the field of view at 30 miles per hour. This is approximately equivalent to a two pixel movement per frame at 500 meters. The intelligent tracker algorithms were tested on data which presented target growth rates of two to six pixels per frame and with 18 pixels target movement between frames.

The test data selected from the NV&EOL 875-line FLIR data base was from Tape #5 11/15/77, tape position 3180. Figure 3-37 shows the scenario. The tank starts from the left side of the road. The helicopter is at long range initially. The helicopter flies over the tank when it is between two APC's.

The processed images were approximately 1200 frames into the run and are characterized in Figure 3-38. Note that in one case, 1316 versus 1330, the difference in image size is 20 percent, which characterizes the target growth rate at ranges closer than 250 meters. The target grows at a rate of 8 pixels per frame as the range closes to 125 meters and with one-quarter of a second to impact.

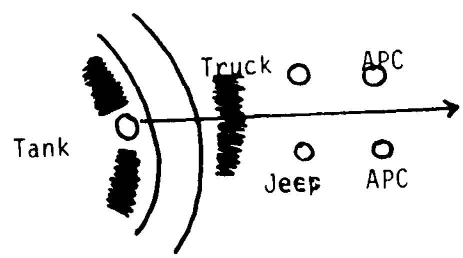


Figure 3-37. Scenario

IMAGE	TANK SIZE (lines high)
1267	24
1288	26
1316	30
1330	36
1344	39

Figure 3-38. Image Characteristics

The images are shown in Figure 3-39 through 3-43. The extracted edges for each of the images are shown in Figures 3-44 through 3-48. Note that the aimpoint is included in Figures 3-39 through 3-43. The aimpoint was placed on image 1267, successive registrations were then placed on it on the following windows. The successive aimpoint placement was achieved by registering an 80 pixel by 50 pixel image containing the target on the following image of 128 pixels by 128 pixels in size. To test the accuracy of the registration results, a point was selected on each tank which appears repeatable. In this case, the point selected was an interior feature which appeared to be the "tread wheel" or sprocket which was approximately 2 pixels by 2 pixels.

Figure 3-39. Image 1267

Figure 3-40. Image 1288

Figure 3-41. Image 1316

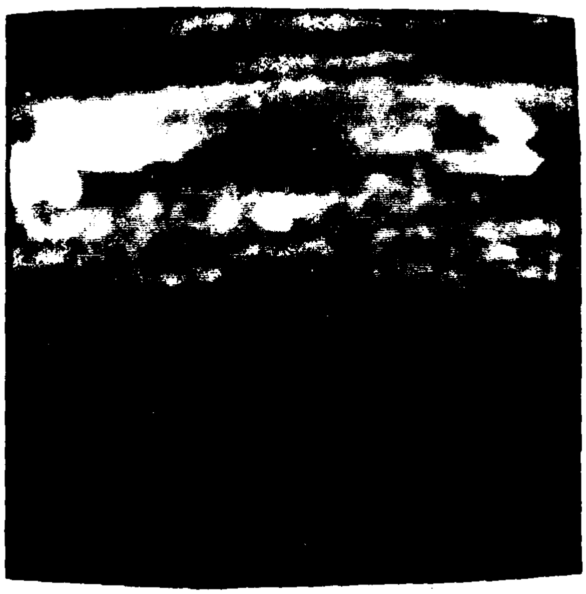


Figure 3-42. Image 1330

Figure 3-43. Image 1344

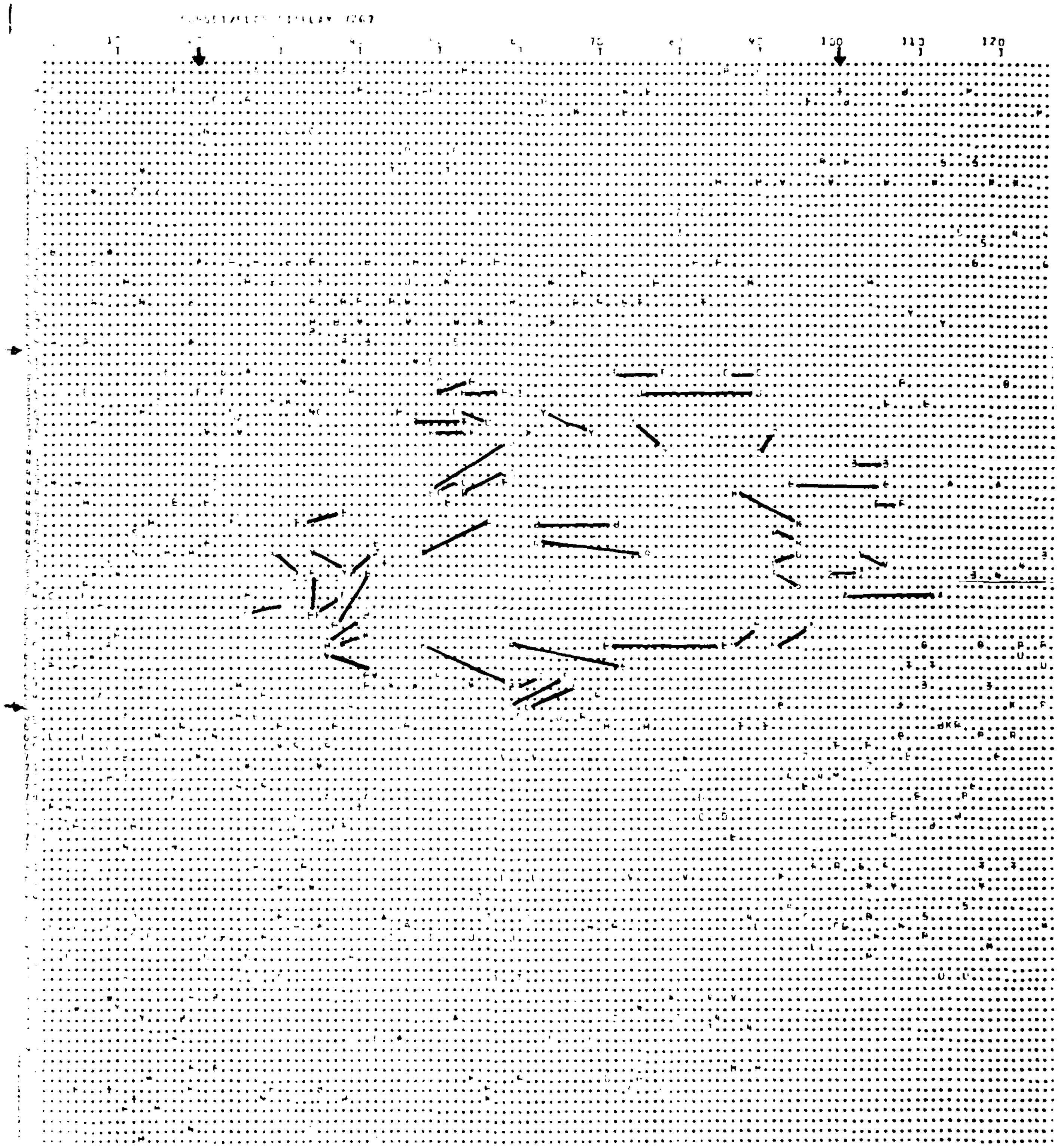


Figure 3-44. Edges for Image 1267

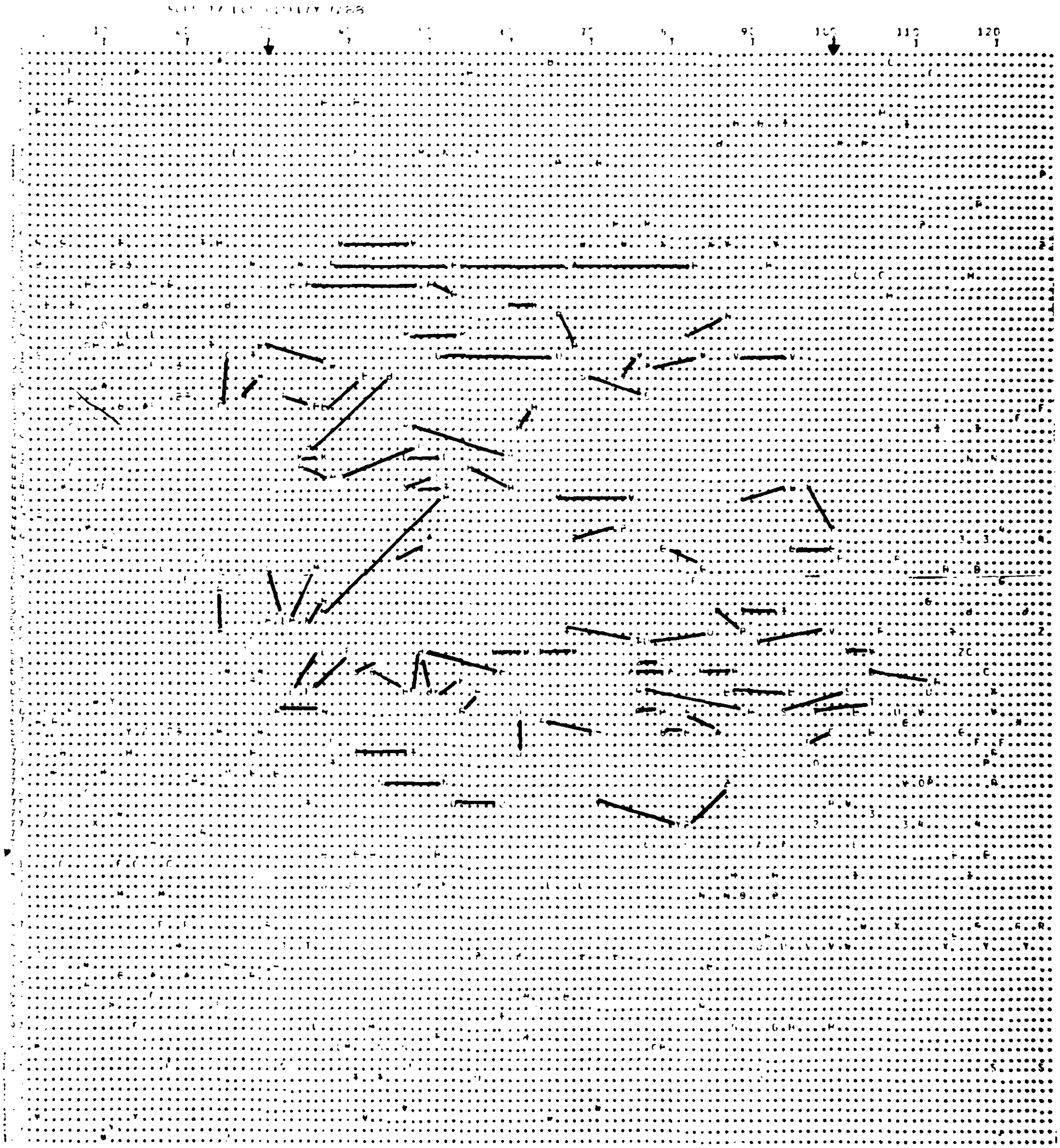


Figure 3-45. Edges for Image 1288

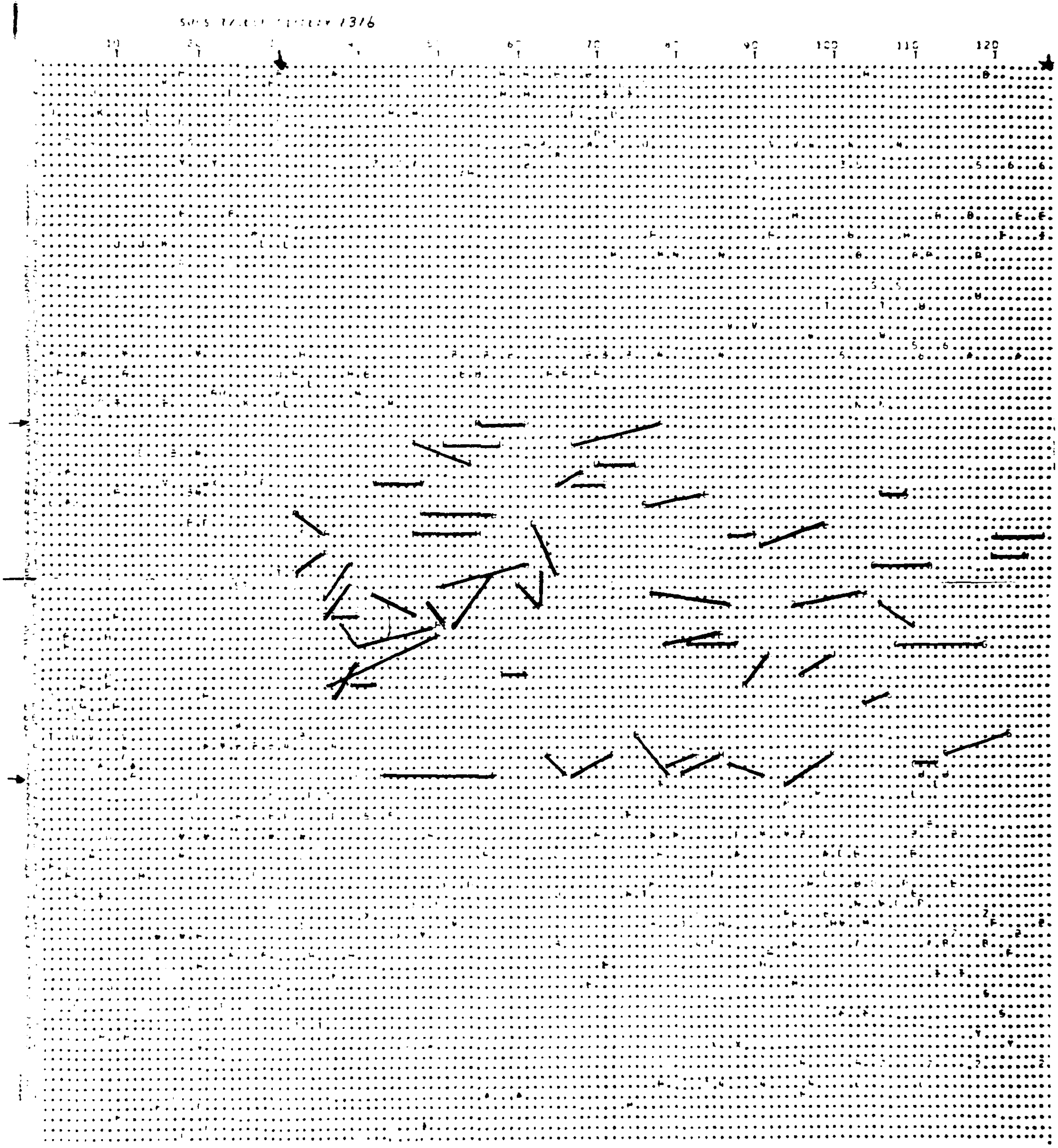


Figure 3-46. Edges for Image 1316

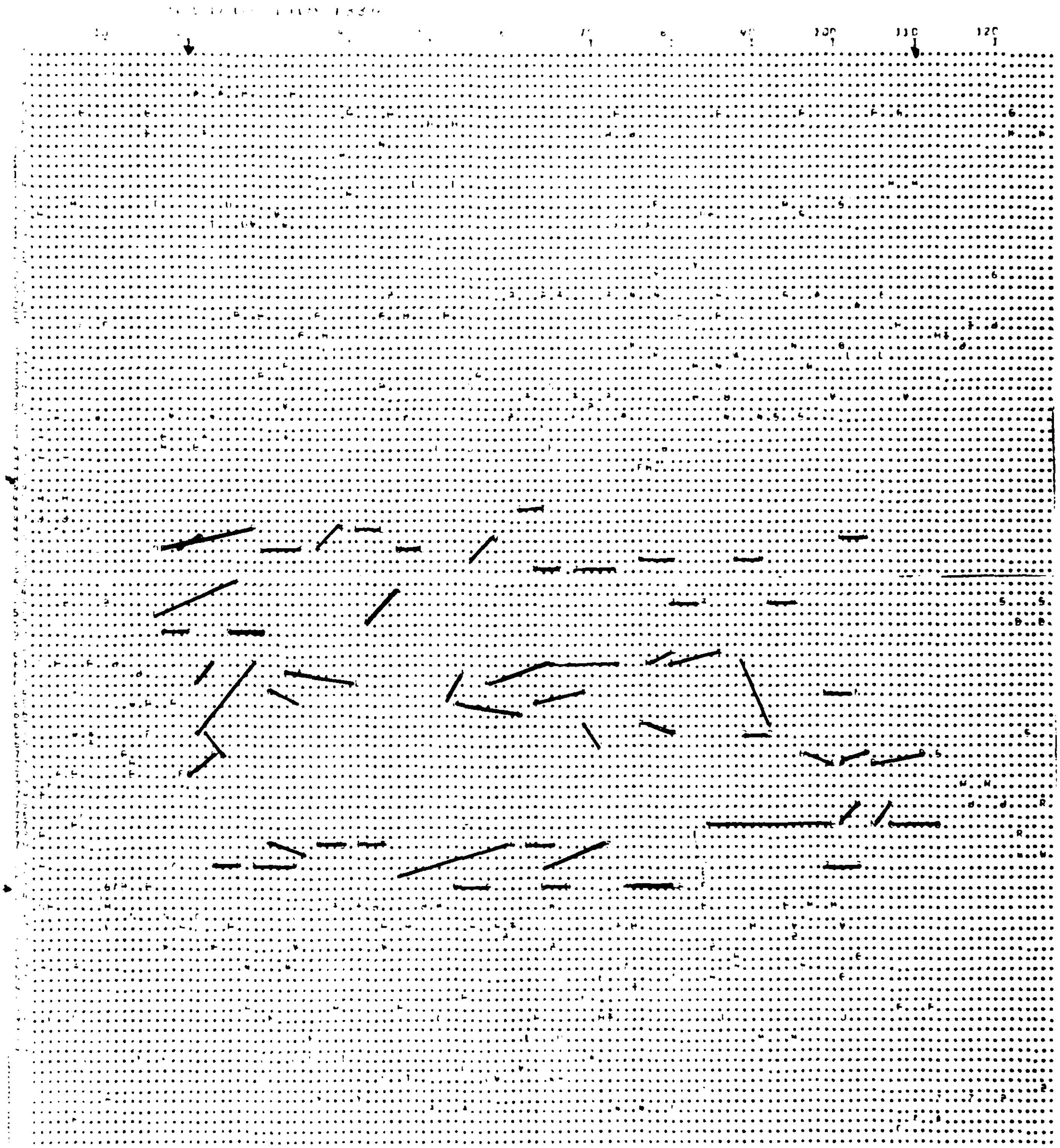


Figure 3-47. Edges for Image 1330

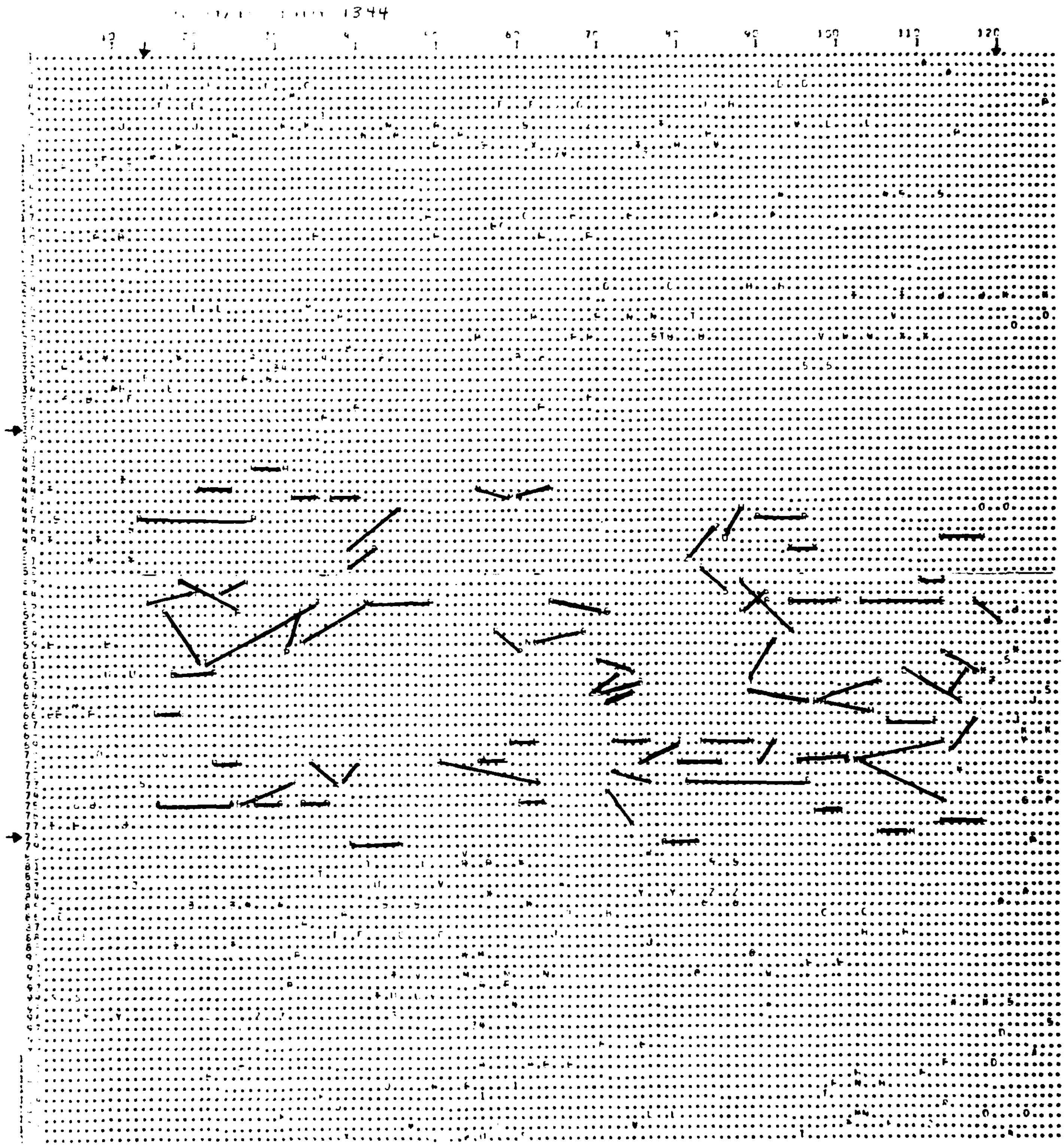


Figure 3-48. Edges for Image 1344

The difference between the sprocket position in the pairs of images is shown in Figure 3-49. The registration results are also shown for the image pairs.

IMAGE PAIRS	BENCH MARK	REGISTRATION	RMS ERROR
1267 vs 1288	-2, +1	-2, 0	1
1288 vs 1316	0, -9	0, -9	0
1316 vs 1330	+14, -9	+13, -10	1.4
1330 vs 1344	-4, -10	-3, -9	1.4

Figure 3-49. Registration Results

In conclusion, the results are in the same accuracy range as the previous results shown in the Fourth Quarterly Report. These accuracies in pixels translate into linear error which is quite small. There is approximately a 0.1 meter per pixel at these ranges, so the linear error is of the order of approximately .1-.2m.

4.0 HARDWARE IMPLEMENTATION

The purpose of this section is to describe the implementation of the intelligent tracker algorithms in very large scale integrated (VLSI) and very high speed integrated (VHSIC) circuit hardware. The first paragraphs, "System Constraints" and "Hardware Constraints", serve as introductions and describe the approach selected. The third paragraph, "Feasibility-An Overview", is a general evaluation of feasibility and it describes the promise of VLSI/VHSIC implementation of the intelligent tracker algorithms. The fourth paragraph, "Parallel Implementation of the Segmentation Algorithms", offers an approach to segmenting an object which lies across the artificial boundaries created by parallel implementation. The fifth paragraph, "Superslice - A Closer Look", examines the algorithm's performance as seen in the intelligent tracker work and suggests a degree of implementation rigor necessary for adequate performance. Also, a Superslice implementation is sketched which uses substantial word widths as contrasted with bit plane and cellular array machines. The final paragraph, "Future Directions", describes hardware implementation efforts to be pursued in the next quarter.

4.1 SYSTEM CONSTRAINTS

In the Second and Third Quarterly Reports, tactical scenarios were developed in which the intelligent tracker played a major role. It is not the purpose of this section to review that work, but rather to select one scenario which presents a greater hardware implementation challenge than some of the others. We will show that the scenario chosen can be realized as a practical system within the time frame of VHSIC and should benefit substantially from the VHSIC program developments.

The Intelligent Tracker found application onboard the helicopter, RPV, and tactical missile (Precision Guided Missile, PGM). The hardware implementation challenge, in terms of volume and power, seems to be the greatest for the small tactical missile. Volume and power constraints of approximately 16 cubic inches and 16 watts have been stated for the 155 mm shell (3 inch diameter) by ARRADCOM personnel. The helicopter, on the other hand, may be assumed to accept requirements for less than 500 cubic inches and 150 watts of power. The RPV has several designs which fall in this range, with the mini-RPV requirements tending toward 100 in^3. The helicopter, in terms of the number of pixels processed per second, will probably require a higher rate than the small tactical missile for which a staring mosaic of perhaps 256 pixels x 256 pixels can be projected. At a frame rate of 30 frames per second, the processing rate for the helicopter is 22.5×10^6 pixels/sec. and 1.97×10^6 pixels/sec. for the small tactical missile. There is roughly an order of magnitude difference in required speed. The real question is whether the small tactical missile volume and power density goals can support the memory, control, overhead, and image processing algorithms necessary to achieve a 2 megapixels/sec. rate. In a larger volume, for example, additional processors can be added to obtain higher speeds with a sometimes minimal increase in overhead. However, in the small tactical missile case, the overhead will represent a higher percentage of the overall volume. An example can be found in the case of a power supply. A fixed volume of equipment is necessary to achieve levels of power below approximately 75 watts. That is, a power supply of 25-75 watts will have the same volume. Above 75 watts, the volume increases linearly with power. In conclusion, the small tactical missile (and artillery shell) probably offers the greatest hardware implementation challenge because the ratio of fixed overhead (control, memory, etc.) processing to ALU (Arithmetic Logic Unit) processing will be higher than for the other systems.

The small tactical missile with image processing is a potent weapon and is the forerunner of the smart artillery shell. Weapons systems based on these concepts appear to have great leverage in redressing the massive numerical advantage enjoyed by the WARSAW PACT and are considered high priority DoD developments.

The small tactical missile scenario selected should be realistic and practical within the 1985 time frame which is the projected availability date for VHSIC chips. There is a hierarchy of scenarios which vary from a missile which tracks a laser spot on the target to the completely autonomous ("fire and forget" or "lock on after launch") artillery shell with a range of 20 km. As we move from the laser-guided missile to the completely autonomous artillery shell the degree of complexity of the sensor-image processing system increases. The laser-guided tactical missile is in the field today; the threat to this system is the increasing vulnerability of the launch helicopter which also provides the laser spot. In the Second Quarterly Report (p. 1-13), a maximum exposure time of 20 seconds was assumed before survivability dropped drastically. A tactic to circumvent this problem is to have another concealed helicopter place the laser beam on the target from a mast mounted sight. The missile is launched from the first helicopter in cooperation with the second. The next step in the hierarchy is to delete the laser and use the same helicopter to designate targets and launch the missiles. This is done by providing image processing for both the helicopter and missiles which registers the scene and the target between the two before launch. The missile sensor resolution is such that the target is seen by the missile before launch. This process is called "lock on before launch" and is described in some detail in the Second and Third Quarterly Reports. The remainder of the steps in the hierarchy are associated with the process of "lock on after launch, (LOAL)" which culminates in the completely autonomous artillery shell. The first step in the LOAL hierarchy finds the helicopter obtaining an image of the target scene in the pop-up mode. The helicopter, initially below the tree tops, pops-up into view, obtains an image of the scene, places it in a frame storage device, and ducks back down. The scene is then transferred to the missile and it is launched from the helicopter while it is still down behind the trees. The launch trajectory takes the missile above the trees and toward the target on a course which may or may not duplicate the perspective and orientation from which the helicopter obtained its reference image. Complex processing routines in the missile register the reference image with the image now seen and set the missile guidance controls accordingly. Further, the missile sensor resolution need not be as good as the helicopter sensor resolution because it can steer using back- ground scene matching until it comes within detection range of the target.

The next step in the hierarchy involves the 20 km smart artillery shell and consists of two phases. The first phase has the reference image provided by an RPV or forward observer. The reference image is transmitted back to the artillery battery and loaded in the shell. This reference image is matched by the smart artillery shell during flight and provides controls to guide the shell into the vicinity of the targets. The scene matching provides midcourse corrections. The shell can be guided by scene matching on the background, and not the target, thus easing the sensor resolution requirements. The scene matching here is more difficult than in the previous case because larger differences in perspective, orientation, and scale may exist between the reference image and the instant image. The second phase is the completely autonomous artillery shell in which the reference image is deleted. Here, the challenge seems to lie in resolving the conflicts among the sensor-image processing constraints representing different parts of the scenario.

In conclusion, it seems that laser-guided scenarios will continue to be affected by survivability considerations. At the other end of the spectrum, a number of technical issues, not associated with VHSIC, need to be resolved with regard to the semi-autonomous and autonomous artillery shells. A compromise seems to be the LOAL scenario in which the sensor and image processor are contained in a small tactical missile (6 inch diameter) rather than an artillary shell with its attendant launch problems. This same scenario restricts the magnitude of the scene matching problem by confining the original mismatch. Here, VLSI/VHSIC technology could be truly viewed as a technical insertion in the post 1985 period. It appears that the image processing state of the art will advance to accommodate the LOAL, small tactical missile.

4.2 HARDWARE CONSTRAINTS

The purpose of this section is to describe the VHSIC hardware constraints placed upon implementation of such systems. In particular, VLSI hardware in the form of gated arrays of approximately 3000 gates per chip and VHSIC hardware in the form of 30,000 gate chips are examined. The VLSI chips are the "glue" chips such as multiplexers and memory address generators which form the connections or "glue" between the VHSIC chips. Consider the VLSI, VHSIC approach to the throughput problem of a digital signal (image) processor. The following four paragraphs are excerpts from Westinghouse work described by reference 2.

The task of improving the processing capability of a signal processor consists largely of finding ways to increase data throughput rates. A most important area to be considered is that of circuit technology, with the aim of minimizing the delay associated with each logic gate. To this end, logic circuitry with the highest speed practical is highly desirable. Another important area is that of circuit integration, which when applied to a greater degree can decrease overall propagation delay. This is achieved by reducing the delay associated with propagation of signals from one integrated circuit to another by substitution of faster, on-chip propagation on a larger LSI circuit. Increasing the degree of circuit integration beyond that of generally available MSI/LSI circuits, however, usually entails a significant additional expense and design cycle time when producing fully custom VLSI circuits. The VLSI gated array at 3000 gates per chip is a good solution to the problems of expense and design cycle time. The last layers of metal deposition define the personality of the gated array from a multiplexer, or a multiplier, to an eight bit arithmetic logic unit. A combination of diffusion and ion implantations define a fixed set of transistors and resistors arranged in three types of cell groupings. Each type of cell is customized by configuring the two-level metal interconnection.

The three cell types are: 1) internal cell logic, 2) input/output cell, and 3) input cell. The internal logic cell consists of 26 transistors and 13 resistors, components sufficient to configure four 3-input OR gates (or other configurations totalling up to 14 inputs) and a number of other circuits. The internal schematic is shown in Figure 4-1. Figure 4-2 is an example of a circuit configured from the internal cell components.

The input/output cell consists of 12 transistors and eight resistors and can be configured as a 3-input gate/buffer for input to the chip or output from the chip. The input cell consists of seven transistors and three resistors and is electrically the same as the input version of the input/output cell. A Westinghouse Emitter Coupled Logic (ECL) gated array is composed of 48 internal cells in a 6x8 matrix, 24 input/output cells and 16 input cells. Pen plots of the overall array and detail of one corner are shown in Figures 4-3 and 4-4. An example of a personalized gated array is shown in the micrograph of Figure 4-5.

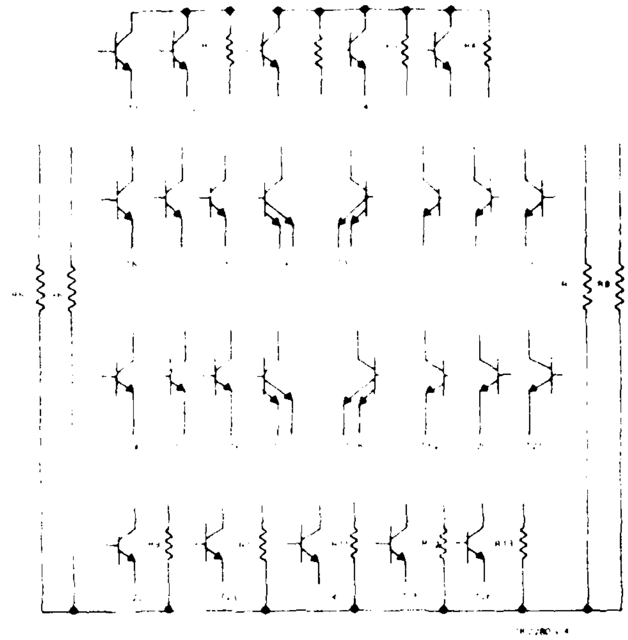


Figure 4-1. Internal Logic Cell Devices

Quad 3 — Input Gate

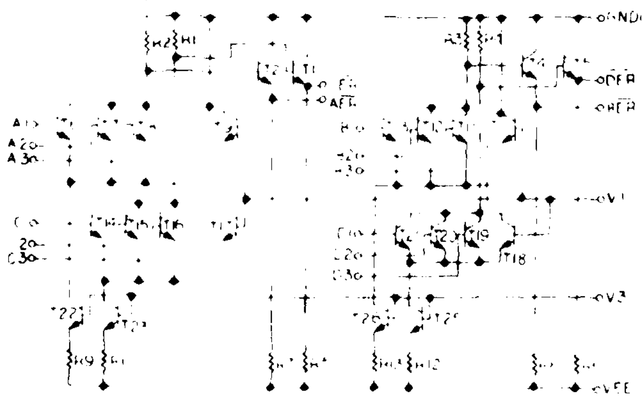


Figure 4-2. Logic Cell Example (Quad 3-Input NOR)

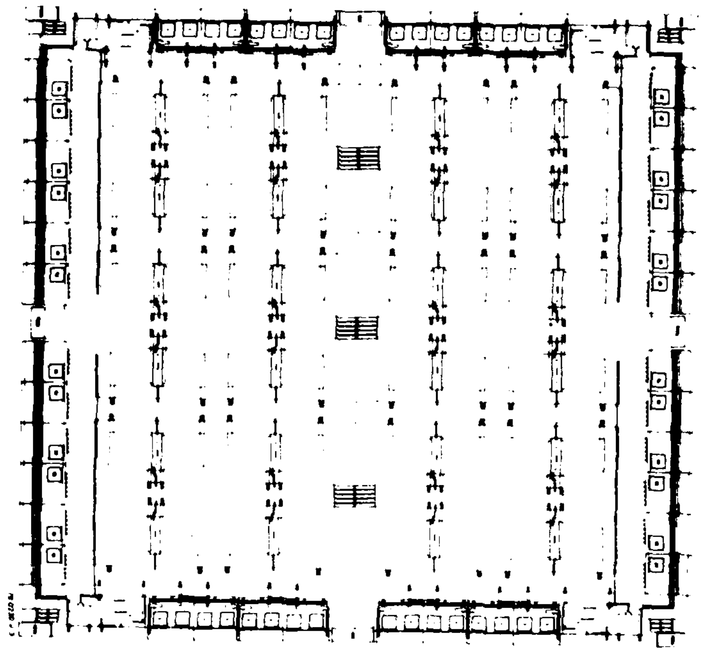


Figure 4-3. Overall Array

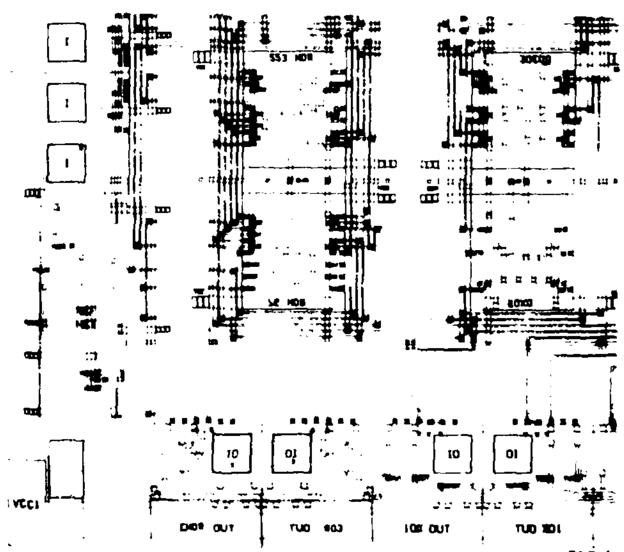


Figure 4-4. Universal Array Corner Detail

While the ECL version has approximately 350 gates on a chip, the CMOS-SOS version has 3000 (VLSI) gates. This latter version can embody the eight bit ALU.

Before considering the CMOS SOS gated array in more detail, consider a review of the procedure necessary to produce a gate array. This will provide a description of the real advantage of a gate array.

The design procedure for a custom VLSI circuit using this array starts with the generation of a logic diagram using conventional CMOS SOS notation. Next a logic partition is done using cell types in the array library. Complete logic simulation of the circuit is done with computer programs of a CAD system, which also includes programs for test generation. Intercell connections between appropriately

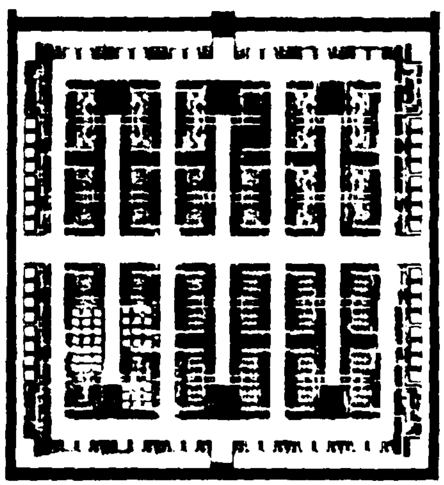


Figure 4-5. Gated Array Micrograph

placed cells are made on a computer-aided graphics system either manually or by means of a computer program for automatic cell placement and interconnect routing, which is also part of the CAD system. The two metal masking levels are subsequently produced for the gate array personal- ization. These patterns are then applied to diffused and metallized wafers to finish the custom integrated circuit.

As mentioned previously one personalization of the CMOS SOS gated array is an eight bit arithmetic logic unit which requires approximately 0.33 watts at a speed of 10 MHz. In a leadless pack configuration, the chip is approximately 1.0 inches x 1.0 inches. The microcode instruction set includes a full complement of logical operations plus addition, subtraction, and 2's complement multiplication. The latter means essentially that 16 clock cycles are necessary to multiply two eight bit numbers. The chip is capable of running faster, but the power requirements increase accordingly. For the purpose of this analysis, it is reasonable to start with the 10 MHz, 0.33 watts combination. The 10 MHz rate means that 10×10^6 microcode instructions can be executed per second. A rough estimate of board spacing indicates that a minimum of 40 such chips can be placed on an 8.0 inch x 10 inch board. An added feature of this personalization is the large quantity of scratch registers above the ALU. It is possible to accommodate 32 scratch registers with approximately 100 leads for this chip which facilitates the implementation, for example, of a 5x5 median filter. The drawback of the personalization is the word width. The dynamic range of future sensors, particularly the reconnaissance line scanners like AAD-5, ESSWACS, and LOREORS, may require more [illegible] bits, which poses additional problems.

[illegible] higher level are the 30,000 gate and upward VHSIC chips which may very [illegible] above gate arrays, not as stand-alone eight bit ALU's, but as ancil- [illegible] a multiplexer. The VHSIC chips will operate in the

25MHz and upward regime and the ALU's will probably be at least 16 bits wide. Further, they will be compatible with higher order languages and architectures such as 1750, NEBULA, and ADA. The projections for the ALU chips are approximately 1.0 inches square and less than 0.6 watts required power, again in a CMOS technology. The VHSIC chip, on the other hand, will not have the versatility of gate array personalizations in reduced design cycle time. However, the repetitive internal cell structure will be easily accommodated on computer aided design (CAD) equipment.

In summary, the hardware characteristics and constraints for VLSI and VHSIC implementations have been described. They will be software programmable and have a very extensive set of arithmetic logic operations. The ALU implementation has been specified because it is a convenient start point. The issue of architecture is not addressed directly in this section; however, paragraph 4.4 and 4.5 deals with some architecture considerations.

4.3 FEASIBILITY - AN OVERVIEW

The purpose of this paragraph is to present a quick look at the VLSI/VHSIC implementation of image processing algorithms, and to indicate feasibility. Certain assumptions are made, based on prior implementations in MSI (medium scale integrated) and CCD (charge-coupled device) circuits.

Based on experience with Auto-Q implementation in MSI at Westinghouse, and additional work in gate arrays and VHSIC with the Maryland-DARPA algorithms, we might assume that the implementation of a sufficient set of image processing algorithms to perform the intelligent tracking functions can be implemented as approximately 1500 microcode instructions. Further, assume that all 1500 instructions must be executed per pixel. These assumptions make no allowance for bandwidth reduction as the image passes through feature extraction, segmentation, and classification. Further, no allowance is made for selectively searching the image nor reducing the image frame processed as close-in tracking is performed.

The instruction set with the above consideration included may decrease by as much as 50 percent. A 50 percent increase is probably not very likely.

From paragraph 4.1, the processing rate for the small tactical missile is 1.9×10^6 pixels per second. Executing 1500 micro instructions per pixel yields a throughput of 2.85×10^9 micro instructions per second. For the VLSI implementation of eight bit wide ALU's, the throughput of each processor is 10×10^6 micro instructions per second. Hence, $.285 \times 10^3$ or 285 ALU's are required to meet the data rate. This simple division assumes some sort of parallel pipeline architecture, which is a brute force attack on the problem. Assume 40 ALU chips can be accomodated on a 8.0 inch x 10.0 inch board, and 80 ALU chips on a board pair. Further, assume there is 0.6 inches between board-pair centers. Then 320 ALU chips can be placed on 4 board pairs for a total volume of 144 cubic inches. This is triple the volume derived for the CCD implementation.[3] However, in the CCD case the design was refined a number of times. In this case, the VLSI implementation appears feasible with a number of very conservative assumptions. The power required is approximately 86 watts which is a factor of three above the required power as described in paragraph 4.1.

For the same parallel pipeline architecture, the VHSIC implementation requires $2850 \times 10^6 / 25 \times 10^6 = 114$ ALU's. Assume that 40 such chips would fit on a 8.0x10.0 inch board. The volume is less than 57 cubic inches with a required power of less than 70 watts.

Consider now a relaxation of one of the conservative assumptions. Assume the intelligent tracker system concept of cueing every 5th frame and tracking in the intervening four frames. For ease of discussion, suppose one portion of the hardware handles cueing, change detection, histogram formation, and close-in tracking. The other performs correlation tracking and aimpoint selection. Further, the former is composed of 1300 instructions and the latter is composed of 200 instructions. Roughly, the first unit requires one fifth of the volume and speed just calculated and the latter requires one seventh. In VLSI, the required volume and power is 48 cubic inches and 28 watts. In VHSIC implementa- tion, the required volume and power are 19 cubic inches and 25 watts. This refine- ment puts the VLSI/VHSIC implementation within range of the small tactical missile requirements. The power supply must also have long shelf life.

In conclusion, a rough calculation based on conservative assumptions and no implementation refinements tends to indicate that VLSI/VHSIC hardware will produce the required volumes and power required by the small tactical missile case and can be software programmable. Smart artillery shells would appear to require phase II VHSIC technology - and, potentially, the development of custom designed VHSIC processor chips.

4.4 PARALLEL ORGANIZATION OF SEGMENTATION ALGORITHMS

To gain speed in performing segmentation algorithms, the concept of a parallel pipeline architecture is often suggested. However, little work has been published in implementing such an organization for segmentation. An approach to the problem is proposed in this section.

The concept of a parallel organization calls for dividing the image into vertical strips and assigning a processor to each strip as shown in Figure 4-6. Each processor is responsible for segmenting its own vertical strip of the image. The problem arises when an object lies across an artificial boundary created by the architecture. This boundary, of course, does not exist in the image. Further, this problem can be overcome in a parallel organization of the median filter, edge extraction, and non-maximum suppression operators. All that is necessary to accommodate a parallel organization for these operators is to overlap the strips

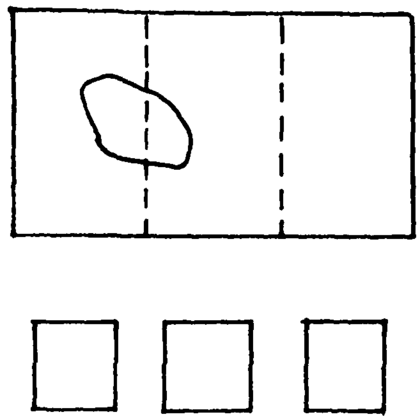


Figure 4-6. Parallel Organization

by one-half the width of the moving window. This allows each processor to access the neighboring pixels in adjacent strips which it needs to process the edge pixels in its strip as the center of a moving window. For the segmentation algorithms the problem is more complicated. As an example consider the Connected Components algorithm described in Appendix A of the First Quarterly Report.

The purpose of the Connected Components algorithm is to segment the image data stream into smaller domains. Each small domain includes a single object in the image plane. This algorithm distinguishes between objects and isolates regions so that classification statistics can be obtained.

Assume that the original image has been thresholded and the result is in binary form with gray levels exceeding the threshold t_i shown as 1's in Figure 4-7. Two image lines are retained in memory so that each pixel can examine its neighbors to the left and right and above and below. No diagonal connections are permitted under this convention, and an adjacent (horizontal or vertical) pixel must be occupied in order to make a connection. No skips or gaps are allowed and the computations start one pixel in from the edge. In

1			1	1	1	1			1	1	
1			1	1	1		1			1	
1			1	1	1		1			1	
1			1	1	1	1	1			1	
1			1	1	1	1	1	1		1	
								1	1	1	

Figure 4-7. Binary Image

Figure 4-8, we have begun the process of labeling the connected components of Figure 4.7. At this point in the process, there are four distinct regions, A, B, C, and D. The only possible connection between regions B and C (to this point) is through a diagonal, which is not allowed. Computations for the fourth row are seen in Figure 4-9. Here there is an equivalence between regions B and C and an equivalent statement, B = C, is carried along. At the end of the sixth row, there is another connection between C and D (C = D) and all the regions are completed as seen in Figure 4-10.

If the above process is repeated, using the partitioned image as shown in Figure 4-6, the result is shown in Figure 4-11. The equivalence will not be discovered. Figure 4-11 shows more sub-regions comprising the objects. Let us assume for the moment that the equivalences within a partitioned strip have been discovered and corrected. The result appears in Figure 4-12.

A			B	B	B	B				D	D
A			B	B	B		C				D
1			1	1	1		1				1
1			1	1	1	1	1				1
1			1	1	1	1	1	1	1		1
									1	1	1

Figure 4-8. Computations for Second Row

A			B	B	B	B				D	D
A			B	B	B		C				D
A			B	B	B		C				D
A			B	B	B	B	C				D
1			1	1	1	1	1	1	1		1
									1	1	1

Figure 4-9. Computations for the Fourth Row

A			B	B	B	B				B	B
A			B	B	B		B				B
A			B	B	B		B				B
A			B	B	B		B				B
A			B	B	B	B	B	B	B		B
									B	B	B

Figure 4-10. Completed Image

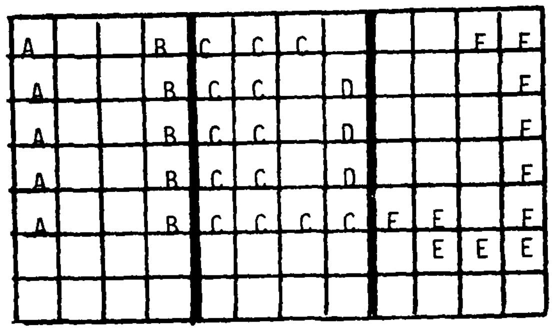


Figure 4-11. Completed Image Without Equivalence with Partitioning

Consider an approach to resolving the dilemma posed by Figure 4-12 which involves a straightforward computation of connected component and equivalence statements and a different memory design. As seen in Figure 4-7, 8, 9, 11 the natural movement of the connected components is from left to right, across an entire line before the next line is analyzed. This is opposite from partitioning the image into vertical strips. However, if there is a directed line connecting adjacent processors from the register stack of one to the other such that the results of processor number 1 are passed to processor number 2, then a line of the image can be executed as shown in Figure 4-13. Processor number 1 is associated with vertical strip number 1, processor number 2 is associated with vertical strip number 2 and so on. In practice, processor number 1 processes line number 1 of vertical strip number 1 and hands it to processor number 2 which processes line number 1 of vertical strip number 2. In the meantime, processor number 1 is now processing line number 2 of vertical strip number 1. In this way, the pipeline is filled such that there is no waiting.

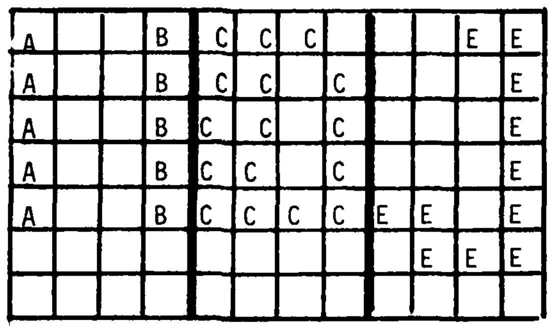


Figure 4-12. Completed Image without Equivalence Across Partitions

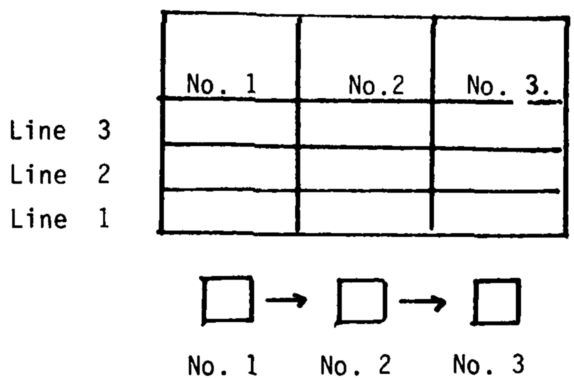


Figure 4-13. Execution for Parallel Organization

In conclusion, a directed line between processors allows the simultaneous horizontal transfer of data between processors. This permits a parallel pipeline organization to avoid the problem of segmenting across artificial boundaries created by the parallel organization. This is a variation on the cellular array architecture for image processing analyzed by Westinghouse - Maryland and sponsored by DARPA - NV&EOL.

4.5 SUPERSLICE - A CLOSER LOOK

The Superslice concept and CCD implementation were developed by Maryland - Westinghouse under sponsorship of DARPA - NV&EOL. In the final report, Ref. 5.3, the threshold selection philosophy was stated on page 4-20 as follows:

> "... The philosophy here is not to attempt to find a single threshold but rather to use a set of thresholds which span the range of gray scale. For the NVL FLIR data, fifteen (15) gray scale levels represented the gray scale ranges and selecting a threshold every three gray scales was deemed to give satisfactory target detections by the University of Maryland."

Although this report was written in March of 1978 and represented the state of the art, cuers are now being held responsible for target detections over a broader and more difficult set of targets. For example, partially occluded targets and target in heavy clutter are notable. What these mean to Superslice from the Intelligent Tracker experience is that the threshold interval of three gray levels is too wide for the expanded target set. In some cases there are only one or two levels at which the target is cleanly segmented and it is either merged or does not appear at all for thresholds on either side of the narrow range of thresholds. Further, aides such as kurtosis (histogram peakedness) and percent of matches above 66% were developed to indicate detections where a edge/perimeter point maximum did not exist.

In conclusion, the VLSI/VHSIC hardware implementation will have to cope with the possibility of a variable number of thresholds for a single frame. However, it may not be necessary to obtain throughput rates commensurate with fifteen thresholds on every frame. For example, one can conceive of utilizing fifteen thresholds in a search mode over several frames if 30 frames per second rates are achieved, i.e. seven or eight thresholds per frame. This is not necessarily required in a track mode. Further, the idea of average power and peak power has application here, because the search mode may require higher clock rates and increased power (peak power) for short periods of time. On the other hand, the track mode may be more appropriate for longer time periods, requiring lower clock rates and lower power. Hence, the average power is within the allowable power density (watts/in^3) for VLSI/VHSIC applications. The Intelligent Tracker concept has the advantage of low average power while the cuer rate can be increased as required.

4.6 CONNECTED COMPONENTS AND WORD WIDTH

The purpose of this section is to broach the idea of implementing bit plane architecture in the traditional ALU. Cellular arrays are also discussed.

Bit plane architecture is defined here as operating on a word by operating on each bit separately. For example, in a traditional ALU, an AND between two eight-bit words results in a one bit flag which describes the result of ANDing the two words. A bit plane implementation results in eight flags each of which signifies the status of each of the eight pairs. If this sort of approach could be applied to the Connected Component algorithm, then it is conceivable that two 16 bit ALU's could perform the segmentation for fifteen superslice thresholds in parallel. Recall that the Connected Components algorithm is performed in binary space. However, it also requires cojnditional branching depending on existence of matching topologically related components. Currently, the hardware implementation calls for sequential execution of successive Connected Component thresholds.

Cellular arrays require an array of processors as dense as the image array itself. Each programmable processor unit must be capable of handling a word width representing the dynamic range of each pixel. Each processor communicates to each of its eight neighbors along directed lines. Although cellular arrays put extraordinary demands on chip fabrication, some of ideas can be adopted and are useful. An example of this is the parallel implementation of segmentation algorithms as described in Section 4.4 of this report.

In conclusion, the bit plane approach appears to be incapable of handling simultaneous branching instructions within a conventional ALU. One concept was adopted from cellular arrays and is described in Section 4.4.

4.7 FUTURE DIRECTIONS

With regard to hardware implemenation, the intent is to continue with the VLSI/VHSIC implementation by removing more of the conservative assumptions.

5.0 REFERENCES

5.1 Edge Point Linking Using Convergent Evidence
D.L. Milgram, page 85, Proceedings DARPA Image
Understanding Workshop, November 1978

5.2 LSI Universal Array Circuits for High Speed
Programmable Processors, Natalie, M.R. &
Brooks, C.W. 1978 NAECON, Dayton, Ohio.

5.3 Algorithms and Hardware Technology for Image
Recognition, Milgram, D.L., Rosenfeld, A.,
Willett, T.J., & Tisdale, G.T. Final Report
under Contract DAAG53-76C-0138, March 31, 1978

DAT

ILM